

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2003-345643

(P2003-345643A)

(43) 公開日 平成15年12月5日 (2003.12.5)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
G 0 6 F 12/00	5 4 5	G 0 6 F 12/00	5 4 5 A 5 B 0 6 5
	5 1 4		5 1 4 E 5 B 0 8 2
3/06	3 0 1	3/06	3 0 1 Z
13/00	5 2 0	13/00	5 2 0 B

審査請求 未請求 請求項の数25 O L (全 30 頁)

(21) 出願番号 特願2002-149977(P2002-149977)

(22) 出願日 平成14年5月24日 (2002.5.24)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 川本 真一

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(72) 発明者 江端 淳

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(74) 代理人 100075096

弁理士 作田 康夫

最終頁に続く

BEST AVAILABLE COPY

(54) 【発明の名称】 ネットワークストレージシステムの仮想一元化方法及び装置

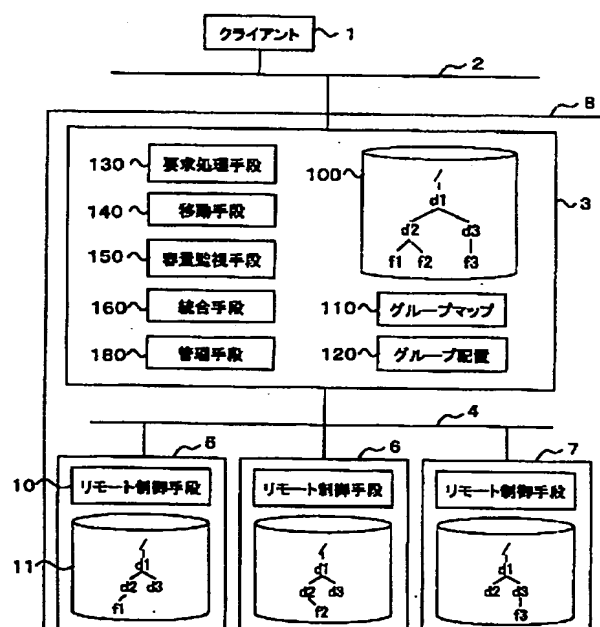
(57) 【要約】

【課題】一つ以上のネットワークストレージを仮想的に一元化するネットワークストレージシステムの仮想一元化方法において、ファイルの格納先ネットワークストレージの情報をファイル数に比例することなくコンパクトに持つ。

【解決手段】 ファイルを一つ以上のファイルグループに分割し、グループ単位で格納先ネットワークストレージを管理する。

【効果】 これまでファイル単位で格納先ネットワークストレージの情報を持つ必要があったが、ファイルグループ単位で格納先ネットワークストレージの情報を持てばよく、保持しなければならない管理情報が減少し、ネットワークストレージの容量を有効に利用できるようになる。また、ネットワークストレージの追加や削除といった構成変更時の格納先ネットワークストレージの情報の書き換えもファイルグループを単位として行えばよいので、書き換えコストを低減できる。

図1



【特許請求の範囲】

【請求項1】 ネットワークに接続された一つ以上のネットワークストレージ装置から構成されるネットワークストレージシステムにおいて、前記ネットワークストレージシステムをクライアントから仮想的に一つのネットワークストレージ装置としてアクセスさせるためのネットワークストレージシステムの仮想一元化方法であって、前記仮想一元化方法によって仮想的に一元化された仮想一元化ネットワークストレージシステムに格納するファイルをグループ単位でネットワークストレージ装置に格納するために、ファイルグループとそのファイルグループに属する全てのファイルの格納先ネットワークストレージ装置との対応を記録するグループ配置表を設け、前記クライアントが前記仮想一元化ネットワークストレージシステムに対してファイルアクセス要求を発行すると、前記ファイルを識別する情報から前記ファイルの属するファイルグループを求め、前記グループ配置表を参照して前記ファイルグループに対応したネットワークストレージ装置を求め、求められたネットワークストレージ装置を前記アクセスの対象ファイルの格納先ネットワークストレージ装置とすることを特徴とするネットワークストレージシステムの仮想一元化方法。

【請求項2】 前記クライアントが前記仮想一元化ネットワークストレージシステムに対してファイル生成要求を発行すると、前記ファイルを識別する情報から前記ファイルの属するファイルグループを求め、前記グループ配置表を参照して前記ファイルグループの格納先ネットワークストレージ装置を求め、求められた格納先ネットワークストレージ装置に前記ファイルを生成することを特徴とする請求項1記載のネットワークストレージシステムの仮想一元化方法。

【請求項3】 前記クライアントが前記仮想一元化ネットワークストレージシステムに対してファイル読み出し要求又は書き込み要求を発行すると、前記ファイルを識別する情報から前記ファイルの属するファイルグループを求め、前記グループ配置表を参照して前記ファイルグループの格納先ネットワークストレージ装置を求め、求められた格納先ネットワークストレージ装置を対象として、ファイルの読み出し又は書き込みを行なうことを特徴とする請求項1記載のネットワークストレージシステムの仮想一元化方法。

【請求項4】 前記クライアントが前記仮想一元化ネットワークストレージシステムに対してディレクトリ生成要求を発行すると、前記仮想一元化ネットワークストレージシステム内の全てのネットワークストレージ装置に対し前記ディレクトリ生成要求を発行し、各ネットワークストレージ装置においてディレクトリを生成することを特徴とする請求項1記載のネットワークストレージシステムの仮想一元化方法。

【請求項5】 ファイル識別子と前記ファイル識別子に

って識別されるファイルが属するファイルグループとの対応関係を与えるグループマップを設け、前記クライアントが前記仮想一元化ネットワークストレージシステムに対し、生成するファイルのパス名を引数としてファイル生成要求を発行すると、前記パス名のファイルに対応するファイル識別子を生成し、前記グループマップを用いて前記ファイル識別子に対応したファイルグループを求め、前記グループ配置表を参照して、前記ファイルグループの格納先ネットワークストレージ装置を求め、前記ネットワークストレージ装置に前記ファイルを生成することを特徴とする請求項1記載のネットワークストレージシステムの仮想一元化方法。

【請求項6】 前記クライアントが前記仮想一元化ネットワークストレージシステムに対し、アクセスするファイルのファイル識別子と対応したハンドルを引数としてファイル読み出し要求又は書き込み要求を発行すると、前記ハンドルからファイル識別子を求め、前記グループマップを用いて前記ファイル識別子と対応するファイルグループを求め、前記グループ配置表を参照して求めたファイルグループの格納先ネットワークストレージ装置を特定し、特定したネットワークストレージ装置に対しファイルの読み出し又は書き込みを行うことを特徴とする請求項5記載のネットワークストレージシステムの仮想一元化方法。

【請求項7】 前記仮想一元化ネットワークストレージシステム内にファイルシステムを設け、前記ファイル識別子は前記ファイルシステムにおけるiノード番号とし、前記グループマップは前記iノード番号を検索鍵としたハッシュ関数とし、前記クライアントが前記仮想一元化ネットワークストレージシステムに対してパス名を指定してファイル生成要求を発行すると、前記ファイルシステムに前記パス名のファイルを生成し、生成したファイルのiノード番号に前記ハッシュ関数を適用してファイルグループを求め、前記グループ配置表を参照して前記ファイルグループの格納先ネットワークストレージ装置を特定し、特定したネットワークストレージ装置に前記ファイルを生成することを特徴とする請求項5記載のネットワークストレージシステムの仮想一元化方法。

【請求項8】 前記クライアントが前記仮想一元化ネットワークストレージシステムに対してアクセスするファイルのハンドルを指定してファイル読み出し要求又はファイル書き込み要求を発行すると、前記ハンドルからiノード番号を求め、前記iノード番号に前記ハッシュ関数を適用してファイルグループを求め、前記グループ配置表を参照して前記ファイルグループの格納先ネットワークストレージ装置を特定し、特定したネットワークストレージ装置に対してファイルの読み出し又は書き込みを行うことを特徴とする請求項5記載のネットワークストレージシステムの仮想一元化方法。

【請求項9】 請求項5記載のネットワークストレージシステム

システムの仮想一元化方法によって仮想的に一元化されたネットワークストレージシステムにおける、ファイルを現在格納しているネットワークストレージ装置から別のネットワークストレージ装置へ移動するファイル移動方法であって、ファイルの移動はファイルグループを単位とし、移動対象ファイルグループと移動先ネットワークストレージ装置を与えると、前記グループ配置表を用いて、前記移動対象ファイルグループの移動前の格納先ネットワークストレージ装置を求め、前記ネットワークストレージ装置に格納されている前記移動対象ファイルグループに属する全てのファイルを前記移動先ネットワークストレージ装置に移動し、前記グループ配置表における前記移動対象ファイルグループに対応するネットワークストレージ装置の記録を、前記移動先ネットワークストレージ装置に設定し直すことを特徴とするファイルの移動方法。

【請求項10】前記仮想一元化ネットワークストレージに格納されているファイルのファイル識別子のそれぞれに対して、前記グループマップを用いてファイル識別子に対応するファイルグループを求め、前記ファイルグループが前記移動対象ファイルグループであれば、前記ファイル識別子で識別されるファイルを移動することを特徴とする請求項9記載のファイルの移動方法。

【請求項11】前記仮想一元化ネットワークストレージシステムにネットワークストレージ装置を追加すると、その追加したネットワークストレージ装置を前記移動先ネットワークストレージ装置に設定し、前記仮想一元化ネットワークストレージシステムの全ファイルグループの中から、一つ以上のファイルグループを選択し、選択したファイルグループの中の一つを前記移動対象ファイルグループに設定してファイルを移動し、全ての選択したファイルグループについて、上記のファイル移動を行なうことを特徴とする請求項9記載のファイルの移動方法。

【請求項12】前記仮想一元化ネットワークストレージシステムにネットワークストレージ装置を追加すると、その追加したネットワークストレージ装置を前記移動先ネットワークストレージ装置に設定し、前記仮想一元化ネットワークストレージシステムのネットワークストレージ装置のうち残容量の最も少ないネットワークストレージ装置に格納されているファイルグループの中から一つのファイルグループを選択し（ステップ1）、前記ファイルグループを移動対象ファイルグループとしてファイルの移動を行ない（ステップ2）、前記ファイルグループ選択とファイルの移動を終了条件が満足されるまで繰り返し行なうことを特徴とする請求項9記載のファイルの移動方法。

【請求項13】前記終了条件は各ネットワークストレージの残容量のうち最大残容量と最小残容量の差が予め定

る請求項12記載のファイルの移動方法。

【請求項14】前記仮想一元化ネットワークストレージシステムからネットワークストレージ装置を削除する際に、前記削除するネットワークストレージ装置に格納されているファイルが属するファイルグループの全てから一つのファイルグループを選択し（ステップ1）、前記ファイルグループを移動対象ファイルグループとし（ステップ2）、前記ネットワークストレージシステム内の前記削除するネットワークストレージ装置以外のネットワークストレージ装置の一つを選択して移動先ネットワークストレージ装置とし（ステップ3）、ファイルの移動を行ない（ステップ4）、前記ステップ1、ステップ2、ステップ3、ステップ4を、前記削除するネットワークストレージ装置にファイルが存在しなくなるまで続けることを特徴とする請求項9記載のファイルの移動方法。

【請求項15】前記ステップ3において、移動先ネットワークストレージ装置として選択するネットワークストレージ装置は、最も残容量の多いネットワークストレージ装置とすることを特徴とする請求項14記載のファイルの移動方法。

【請求項16】前記仮想一元化ネットワークストレージシステムを構成するネットワークストレージ装置の残容量が予め定められた第一の閾値を下回ると、前記ネットワークストレージ装置に格納されているファイルの属するファイルグループの中から一つのファイルグループを選択して移動対象ファイルグループとし（ステップ1）、前記ネットワークストレージ装置以外のネットワークストレージ装置の一つを選択して移動先ネットワークストレージ装置とし（ステップ2）、ファイルを移動し（ステップ3）、前記移動対象ファイルグループが移動前に格納されていたネットワークストレージ装置の残容量が第二の閾値を超えるまで、前記ステップ1、ステップ2、ステップ3を繰り返すことを特徴とする請求項9記載のファイルの移動方法。

【請求項17】前記仮想一元化ネットワークストレージシステムを構成する各ネットワークストレージ装置の残容量の最大残容量と最小残容量の差が予め定められた第一の閾値を上回ると、前記ネットワークストレージ装置に格納されているファイルの属するファイルグループの中から一つのファイルグループを選択して移動対象ファイルグループとし（ステップ1）、前記ネットワークストレージ装置以外のネットワークストレージ装置の一つを選択して移動先ネットワークストレージ装置とし（ステップ2）、ファイルを移動し（ステップ3）、前記ネットワークストレージ装置の残容量の差が第二の閾値を下回るまで、前記ステップ1、ステップ2、ステップ3を繰り返すことを特徴とする請求項9記載のファイルの移動方法。

【請求項18】前記ステップ2において、選択する移動

いネットワークストレージ装置であることを特徴とする請求項１６、１７記載のファイルの移動方法。

【請求項１９】一つ以上のネットワークストレージ装置をクライアントから仮想的に一つのネットワークストレージシステムとしてアクセスさせるためのネットワークストレージ装置の仮想一元化装置であって、ファイル識別子と前記ファイル識別子によって識別されるファイルが属するファイルグループとの対応関係を与えるグループマップと、ファイルグループと前記ファイルグループの格納先ネットワークストレージ装置との対応を記録するグループ配置表と、前記クライアントからの要求を処理する要求処理手段を備え、前記クライアントが前記仮想一元化ネットワークストレージシステムに対してファイルアクセス要求を発行すると、前記要求処理手段は前記アクセス対象ファイルのファイル識別子を求め、前記グループマップを用いて前記ファイル識別子と対応したファイルグループを求め、前記グループ配置を参照して前記ファイルグループの格納先ネットワークストレージ装置を求め、前記ネットワークストレージ装置に対して前記クライアントが発行したファイルアクセス要求を発行することを特徴とするネットワークストレージ装置の仮想一元化装置。

【請求項２０】請求項１９記載のネットワークストレージ装置の仮想一元化装置であって、前記一元管理装置によって仮想一元化されたネットワークストレージシステム内のネットワークストレージ装置に格納されたファイルを別のネットワークストレージ装置に移動するファイル移動手段を更に備え、ファイルの移動が必要な場合に、前記ファイル移動手段が、グループ単位でファイルを移動することを特徴とするネットワークストレージ装置の仮想一元化装置。

【請求項２１】請求項１９記載のネットワークストレージ装置の仮想一元化装置であって、ファイルを格納する手段と、ネットワーク接続されたクライアントから前記ファイル格納手段をネットワークストレージ装置としてアクセスさせるためのネットワークストレージ制御手段を更に備えることを特徴とするネットワークストレージ機能付き仮想一元化装置。

【請求項２２】請求項２１記載のネットワークストレージ機能付き仮想一元化装置であって、クライアントが前記ネットワークストレージ機能付き仮想一元化装置一台と０台以上のネットワークストレージ装置から構成される仮想一元化ネットワークストレージシステムに対してファイルアクセス要求を発行すると、前記ネットワークストレージ機能付き仮想一元化装置の要求処理手段は、前記ファイルアクセス要求のアクセス対象ファイルを格納するネットワークストレージ装置が自身である場合、自身のネットワークストレージ制御手段に対して前記クライアントが発行したファイルアクセス要求を発行して

ネットワークストレージ機能付き仮想一元化装置。

【請求項２３】クライアントのファイルが格納された既設ネットワークストレージ装置が接続されているネットワークに、請求項１記載の仮想一元化ネットワークストレージシステムの仮想一元化方法によって一元化され、一台以上のネットワークストレージ装置から構成されるファイルが一つも格納されていない仮想一元化ネットワークストレージシステムを接続し、前記ファイルを保持したまま、前記既設ネットワークストレージ装置を前記仮想一元化ネットワークストレージシステム内の一つのネットワークストレージ装置として取り込むための、既設ネットワークストレージ装置の仮想一元化ネットワークストレージシステムへの統合方法であって、ファイルグループのそれぞれに対して、前記既設ネットワークストレージ装置及び追加したネットワークストレージ装置の中の一つを対応付けてグループ配置を設定し、前記ファイルが格納されている既設ネットワークストレージ装置の全ファイルに対して新たにファイル識別子を生成し、前記ファイル識別子の全てに対して前記グループ配置を参照し前記ファイル識別子と対応したネットワークストレージ装置を求め、前記ネットワークストレージ装置が、前記既設ネットワークストレージ装置でない場合は、前記ファイルを前記既設ネットワークストレージ装置から前記ファイルグループの格納先ネットワークストレージ装置に移動することを特徴とする既設ネットワークストレージ装置の仮想一元化ネットワークストレージシステムへの統合方法。

【請求項２４】前記グループ配置において、全てのファイルグループの格納先ネットワークストレージ装置を前記既設ネットワークストレージ装置となるように設定することを特徴とする請求項２３記載の既設ネットワークストレージ装置の仮想一元化ネットワークストレージシステムへの統合方法。

【請求項２５】請求項１～９のいずれかに記載のネットワークストレージの仮想一元化方法によって仮想的に一元化された仮想一元化ネットワークストレージシステムにおいて、各ネットワークストレージの残容量を監視し、残容量の合計が予め定められた閾値より小さくなった場合に、管理者に対してネットワークストレージを追加するようメッセージを出すことを特徴とする容量監視手段。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、複数のネットワークストレージ装置をクライアントから一つのネットワークストレージ装置として見せる、ネットワークストレージの仮想一元化方法、及び、仮想一元化装置に関する。

【０００２】

【従来の技術】従来の情報システムは情報を処理する計算機に情報を格納するストレージ装置と、ストレージ装置と

Copyright © 2004 John Wiley & Sons, Ltd.

【0003】

[0004]

1. The first part of the report

るファイルグループを予めグループマップとして定義しておく。グループマップは、例えばファイル識別子を検索鍵としたハッシュ関数として与える。さらに、ファイルグループと、そのファイルグループに属するファイルの格納先ネットワークストレージを定めるグループ配置を設けておく。グループ配置は、例えばファイルグループとネットワークストレージとの対応を記録するテーブルやデータベースとして与える。クライアントがファイルにアクセスする際は、まずアクセス対象ファイルのファイル識別子とグループマップとから、そのファイルが属するファイルグループを求める。次に、グループ配置を参照して求めたファイルグループに対応したネットワークストレージを求める。求めたネットワークストレージがアクセス対象ファイルの格納先ネットワークストレージであるから、前記ネットワークストレージに対してファイルアクセスを行なう。この発明によれば、一つ以上のファイルから構成されるファイルグループを単位として格納先ネットワークストレージの情報を持つため、ファイル毎に管理情報を持つ場合に比べて管理情報が少なくなる。また、仮想一元化ネットワークストレージシステムの構成変更時には、ファイルグループを単位としてファイルの格納先ネットワークストレージを移動する。この際に、グループ配置の移動するファイルグループの格納先ネットワークストレージを変更するだけで良いので、ファイル毎に格納先情報を管理する場合に比べてオーバーヘッドを小さくできる。

【0005】

【発明の実施の形態】＜実施例1＞図1は本発明の第一の実施例を含む情報システムの全体構成を示す図である。クライアント1はネットワーク2を介して本発明の仮想一元化装置3に接続されている。仮想一元化装置3はネットワーク4を介して、ネットワークストレージ5、6、7と接続されている。ネットワーク4を用意する代わりにネットワークストレージ5、6、7をネットワーク2に接続しても良い。ネットワーク2と4を独立すると、アクセス集中時に高い性能を出すことができる。一方ネットワーク2にネットワークストレージ5、6、7を接続すると、ネットワーク4が不要となり、コストを抑えることができる。仮想一元化装置3はネットワークストレージ5、6、7を仮想的に一元化し、クライアント1から一つのネットワークストレージとしてアクセスできる仮想一元化ネットワークストレージシステム8を提供する。

(ネットワークストレージ)ネットワークストレージ5、6、7は公知の技術によって構成されるもので、リモート制御手段10とストレージ装置11とを備える。ストレージ装置11にはファイルシステムを搭載する。ネットワークストレージ5、6、7は専用装置であっても良いし、ストレージ装置を備えた汎用のサーバ、ワークス

ても良い。リモート制御手段10は、ネットワーク上に接続されたクライアントからのマウント要求とファイルの生成、読み出し、書き込み、あるいは、ディレクトリの生成等の要求を受け、ローカルストレージ装置のファイルシステムに対してファイルの生成、読み出し、書き込み、あるいは、ディレクトリの生成等を行なう。リモート制御手段10は、サン・マイクロシステムズ社(Sun Microsystems, Inc.)によって開発されたNFS(Network File System)や、マイクロソフト社(Microsoft Corporation)によって開発されたCIFS(Common Internet File System)等のファイルアクセスプロトコルを使用して、クライアントとの通信を行なう。本実施例1ではファイルアクセスプロトコルとしてNFSを用いるものとする。NFSの要求を受けファイルアクセス処理を行なうリモート制御手段10としては、公知のmountdとnfsdを使用する。

(仮想一元化装置)本実施例の仮想一元化装置3は、ファイルシステム100、グループマップ110、グループ配置120、要求処理手段130、移動手段140、容量監視手段150、統合手段160、管理手段180から構成される。ファイルシステム100、グループマップ110、グループ配置120は主にファイルの格納先ネットワークストレージを定めるために用いられる。詳細は後述の(ファイルシステム)、(グループマップ)、(グループ配置)の項で説明する。要求処理手段130はクライアントからのファイルアクセス要求を処理する。詳細は後述の(要求処理手段)の項で説明する。移動手段140は、ネットワークストレージを追加したり削除したり、あるいはネットワークストレージの一つの残容量が定められた値を下回った場合に、ファイルをネットワークストレージ間で移動する。詳細は後述の(移動手段)の項で説明する。容量監視手段150は各ネットワークストレージの容量を常時監視し、必要があれば移動手段を起動する。詳細は後述の(容量監視手段)の項で説明する。統合手段160は既に運用中の既設ネットワークストレージ装置を仮想一元化ネットワークストレージシステムに統合する際の処理を行なう。詳細は後述の(統合手段)の項で説明する。管理手段180は仮想一元化ネットワークストレージシステム8を管理する管理者からの指示を受け付け、仮想一元化装置3の設定を変更したり、移動手段140や統合手段160を呼び出して構成変更等の要求に対応する。管理者は、クライアント1上の汎用Webブラウザをユーザインタフェースとして管理手段180へのアクセスする。あるいは、クライアント1に専用のユーザインタフェースプログラムを搭載してそれを使用したり、仮想一元化装置にコンソールを接続しそれを用いても良い。仮想一元化装置3の処理の概要を説明する。仮想一元化装置3は、仮想一元化装置

に対してマウント要求を発行して、仮想一元化ネットワークストレージシステム8が共有を許可しているディレクトリをクライアント1のファイルシステムの特定の場所にマウントする。そしてクライアント1がファイルアクセス要求を発行すると、仮想一元化装置3の要求処理手段130がそれを受け、ファイルシステム100、グループマップ110、グループ配置120を用いて、アクセス対象ファイルの格納先ネットワークストレージを求める。そして、その格納先ネットワークストレージに対してクライアントから受けたファイルアクセス要求を送る。格納先ネットワークストレージがファイルアクセスを実行し終わると、仮想一元化装置3は格納先ネットワークストレージからの得られたデータを要求元クライアント1に返す。このように、クライアント1はアクセス対象ファイルがどのネットワークストレージに存在するか知る必要はなく、複数のネットワークストレージを一つの大きなネットワークストレージとしてアクセスできる。なお、本発明の仮想一元化装置3は、全てのファイルの一つ以上のファイルグループに分割し、ファイルグループごとにファイルの格納先ネットワークストレージ装置を与え、同一ファイルグループに属するファイルは同じネットワークストレージに格納する。ファイルグループは、使用するネットワークストレージ5、6、7の数と同数があるいはそれより多く設ける。ファイルグループに属するファイルの容量が、ファイルグループ毎に大きく異なると、それを格納するネットワークストレージ間で使用容量のアンバランスが生じる。これを改善するため、ファイルグループは、ネットワークストレージ数の10倍から1000倍程度設ける。こうすることで、ファイルグループ当りの容量のばらつきを減少し、ネットワークストレージ間の使用容量のアンバランスを解消する。このようにファイルグループ単位で格納先ネットワークストレージを定めることで、従来技術で発生するファイル数に比例して管理情報が増加しストレージを圧迫する問題を解消でき、また、ファイル移動時に各ファイル毎に格納先ネットワークストレージの情報を書き換える必要がなくなり、オーバーヘッドを減少できる。以下では仮想一元化装置3の構成要素の詳細について説明する。

(ファイルシステム)ファイルシステムは、仮想一元化ネットワークストレージシステム全体のディレクトリ構造と、全ファイルのファイル識別子を管理する。仮想一元化装置3が使用するファイルシステム100はオペレーティングシステムが提供するファイルシステムを使用する。図2はサン・マイクロシステムズ社のSolaris、インターナショナル・ビジネス・マシーンズ社(International Business Machines Corporation)社のAIX、ヒューレット・パッカード社(Hewlett-Packard)社のVMS等のUNIX(登録商標)系オペレーティングシステムにおけるファイルシステムの基本構造を示したものである。図2のファイルシステムは図3のディレクトリ構造を持つファイル郡を格納した様子を

示している。図3の/、d1、d2、d3はディレクトリであり、f1、f2、f3はファイルを示している。/はルートと呼ばれディレクトリの始点を表す。ファイルとディレクトリをまとめてファイルオブジェクトと呼ぶ。ファイルオブジェクトはファイルシステム内でiノードと呼ばれるデータ構造1001~1007として記録される。iノードはそれを識別するiノード番号という固有の識別子を持つ。図2の例では、/ディレクトリにはiノード1001が対応しそのiノード番号は2番である。ディレクトリd1にはiノード1002が対応しそのiノード番号は5番である。ディレクトリd2にはiノード1003が対応しそのiノード番号は7番である。ディレクトリd3にはiノード1004が対応しそのiノード番号は9番である。ファイルf1にはiノード1005が対応しそのiノード番号は10番である。ファイルf2にはiノード1006が対応しそのiノード番号は14番である。ファイルf3にはiノード1007が対応しそのiノード番号は16番である。ファイルシステムにおいて、ファイルオブジェクトの識別にはiノード番号が使用される。一方ファイルオブジェクトの名前や配置場所を管理するディレクトリ情報は、ディレクトリエントリという構造1008、1009、1010、1011によって管理される。ファイルシステム内のディレクトリのそれぞれに対して一つディレクトリエントリが設けられる。ディレクトリ/、d1、d2、d3に対してそれぞれディレクトリエントリ1008、1009、1010、1011が対応する。ディレクトリのiノードは、そのディレクトリに対応するディレクトリエントリへのポインタを持つ。例えば図2の場合、ディレクトリd1に相当するiノード1002はディレクトリd1に対応するディレクトリエントリ1009へのポインタを保持する。ディレクトリエントリは自身のディレクトリのiノード番号とそのディレクトリ名(.で表される)の組、親ディレクトリのiノード番号とそのディレクトリ名(.で表される)の組、及び、自身のディレクトリの直下にあるファイルまたはディレクトリのiノード番号とファイル名又はディレクトリ名の組を構成要素とする。ディレクトリ構造は、iノードとディレクトリエントリの相互チェーンによって管理される。例えば、/ディレクトリに相当するiノード1001はディレクトリエントリ1008を示しており、このディレクトリエントリを参照すると、/ディレクトリの直下にd1という名前のファイルオブジェクトが存在しそのiノード番号が5番であることが分かる。次に、5番のiノード1002を参照すると、ファイルオブジェクトがデ

クトリエントリ1009を参照することによって、ディレクトリd1の直下にiノード番号7番のd2という名前のファイルオブジェクトとiノード番号9番のd3という名前のファイルオブジェクトが存在することがわかる。さらに、iノード番号9番のiノード1004を参照すると、それがディレクトリであることがわかり、指し示すディレクトリエントリ1011を参照することによって、直下にiノード番号16番のf3という名前のファイルオブジェクトが存在することがわかる。このようにして、iノードとディレクトリエントリによってディレクトリ構造が記録される。次に、ファイルに対応するiノード1005、1006、1007はファイルの中身であるデータを格納するデータブロックへのポインタを記録する。図2の場合、ファイルf1に対応したiノード1005はデータブロック1012を一つ指しており、このデータブロック1012内にファイルf1のデータが格納されている。複数のブロックを使用するファイルの場合、iノードはそれらすべてのブロックへのポインタを記録する。以上、UNIX系オペレーティングシステムが提供するファイルシステムの一般的構造について説明した。通常ファイルシステムは、ファイルのディレクトリ構造や識別子を管理すると共に、ファイルの実体であるデータを保管する。しかし、本発明の仮想一元化装置3は、ファイルシステム100を、ファイルのディレクトリ構造とファイル識別子(以後管理情報と呼ぶ)の管理にのみ使用し、ファイルの実体であるデータは各ネットワークストレージ5、6、7に格納する。このため、仮想一元化装置3はファイルシステム100に対して容量0のダミーファイルを生成する。クライアントが仮想一元化ネットワークストレージシステム8に対して図3のようなディレクトリ構造を持つファイル郡を格納した場合、仮想一元化装置3はファイルシステム100に図2の構造を作るが、ファイルの実体であるデータブロック1012、1013、1014は使用しない。また、仮想一元化装置3はファイル識別子としてiノード番号を利用する。なお、ディレクトリ構造とファイル識別子の管理ができれば必ずしもファイルシステム100を使用する必要はない。ファイル識別子が全ファイルに対して一意であることを保証しつつ、データベースを利用して管理しても良いし、専用のテーブルを設けて権利しても良い。

(グループマップ)グループマップ110は、ファイルとそのファイルが属するファイルグループとの対応を与える。具体的にはファイルの識別子であるiノード番号と、そのiノード番号で識別されるファイルの属するファイルグループの識別子を与える。グループマップ110の具体例として、ファイルのiノード番号1を検索鍵としハッシュ値をファイルグループの識別子Gとするハッシュ関数

によって与える。例えばファイルをN個のファイルグループに分割する場合のハッシュ関数として、iノード番号1にファイルグループ数Nのモジュロ演算(Nで割った余り)を適用して得られた値をファイルグループの識別子とする。

【0006】

$G = \text{Hash}(I) = I \bmod N$

0~8までの9つのファイルグループを設けた場合のグループマップのハッシュ関数は以下ようになる。

【0007】 $\text{Hash}(I) = I \bmod 9 \dots \textcircled{1}$

ハッシュ関数は上記のモジュロ演算の他、iノード番号Iに対して一意にファイルグループを対応付けることができれば何を使用しても良い。ただし、特定のファイルグループに属するファイルの数が他のファイルグループに属するファイルの数より極端に多くなるようなハッシュ関数を使用すると、それらのファイルグループを格納するネットワークストレージ間で容量のアンバランスが生じる。この問題を発生させないようにするには、なるべく各ファイルグループに均等にファイルがばら撒かれるような関数を選ぶべきである。

(グループ配置)グループ配置120の構造を図4に示す。

グループ配置は、ファイルグループと、そのファイルグループに属するファイルの格納先ネットワークストレージとを対応付ける役割を持つテーブルで、ストレージ装置内に格納される。1201の行は9つのファイルグループの識別番号0~8を示しており、1202の行は対応する1201行のファイルグループのファイルの格納先ネットワークストレージの識別番号を示す。図4の例で、ファイルグループ1に属するファイルの格納先はネットワークストレージ1であり、ファイルグループ6に属するファイルの格納先はネットワークストレージ3である。仮想一元化装置3は起動時に、グループ配置の1202の行に、各ネットワークストレージの識別子がほぼ均等に出現するように、初期化を行なう。グループ配置120は、ネットワークストレージの追加、削除、平滑化、及び、既存ネットワークストレージの取り込みの際に書き換えられる。なお、グループ配置120は、ファイルグループと格納先ネットワークストレージとの対応が取れ、かつ、仮想一元化装置3が何らかの障害によってダウンしてしまっても、対応関係が消失してしまわなければ、データベースを利用して良いし、その他どんな方法であっても良い。なお、グループ配置120は、ファイルアクセスの際に必ず参照するため、グループ配置がストレージ内に格納されていると、性能低下を招く。このため、ストレージ内に格納されたグループ配置120のコピーを高速にアクセス可能なメモリに配置しても良い。この場合、グループ配置を書き換える際は、メモリとストレージとを両方とも書き換える必要がある。

(管理情報)管理情報500は、仮想一元化装置3が管理するファイルシステム100の管理情報である。

のファイルシステムのマウント要求とファイルアクセス要求を処理する。本実施例ではクライアント1と要求処理手段130とは、前述のNFSプロトコルを用いてファイルアクセスのやり取りを行なうものとする。一元管理装置3とネットワークストレージ5、6、7の間もNFSプロトコルを用いてファイルアクセスのやり取りを行う。なお、クライアント1がCIFSにのみ対応している場合、一元管理装置3の要求処理手段130はクライアント1とCIFSにてやり取りを行う必要がある。これを実現するため、一元管理装置3はネットワークストレージ5、6、7とはNFSでやり取りを行い、クライアント1とだけCIFSでやり取りを行う。CIFSクライアントとのやり取りは、Sambaなどの公知な技術を用いることで実現できる。NFSではまずクライアントがサーバに対してマウント要求を発行して、サーバ側の共有ファイルのルートディレクトリ(マウントポイントと呼ぶ)をクライアントのファイルシステムの特定の場所にマウントする。要求処理手段130はクライアントからのマウント要求に対して、公知のmountdを利用してマウント要求に答える。mountdは、仮想一元化ネットワークストレージシステム8のマウントポイントのファイルハンドル(後述)をクライアント1に返す。NFSプロトコルに規定されたファイルやディレクトリに対するアクセス要求には、ファイルの生成を行なうCREATE要求、ファイル名を与えてファイルハンドルを取得するLOOKUP要求、ファイルを読み出すREAD要求、ファイルを書き込むWRITE要求、ファイルの属性を設定するSETATTR要求、ファイルの属性を読み出すGETATTR要求、ファイルを削除するREMOVE要求、ファイルの名前を変更するRENAME要求、ディレクトリを生成するMKDIR要求、ディレクトリを読み出すREADDIR要求、ディレクトリを削除するRMDIR要求がある。以下各要求毎に要求処理手段3の動作を説明する。

(CREATE要求の処理)クライアント1はファイルを生成するディレクトリのファイルハンドルHと、生成するファイルの名前Fを引数としてCREATE要求を発行する。ファイルハンドルは、クライアント1が後に説明するLOOKUP要求等を発行すると返回值として得られるファイルを識別するデータ構造である。ファイルハンドルの構成を図6に示す。ファイルハンドル170は親ディレクトリのiノード番号フィールド1701と、ファイルオブジェクトのiノード番号フィールド1702を含んでいる。要求処理手段130がクライアント1からCREATE要求を受けると、図5の1301を呼び出し、要求を処理する。まず1302において、親ディレクトリのファイルハンドルHのiノード番号フィールド1702から、ファイルを生成する親ディレクトリのiノード番号P1を取り出し、このiノード番号で

ミーファイルを生成する。生成したファイルのiノード番号をIとする。次に1303において、生成したファイルのiノード番号Iに対して上記(ファイルマップ)の項で説明したハッシュ関数Hashを適用し、ファイルグループGを求める。次に、1304において上記(グループ配置)の項で説明したグループ配置120を参照し、ファイルグループGに対応するネットワークストレージNを求める。求めたネットワークストレージNがクライアント1が発行したCREATE要求の対象ファイルの格納先ネットワークストレージとなる。そして、1305において、クライアント1から送られてきたCREATE要求をネットワークストレージNに対して発行する。最後に1306で、ネットワークストレージ装置NからCREATE要求の応答が返ると、ファイルシステム100上でダミーファイルを生成したディレクトリのiノード番号P1と生成したダミーファイルそのもののiノード番号Iからファイルハンドルを構成して、それを要求元クライアント1に返して処理を終了する。

(LOOKUP要求の処理)クライアント1はファイルハンドルを取得する対象ファイルオブジェクトの名前Fと、そのファイルオブジェクトの親ディレクトリのファイルハンドルHを引数としてLOOKUP要求を発行する。要求処理手段130がクライアント1からLOOKUP要求を受けると、図7の1311を呼び出す。まず1312において、親ディレクトリのファイルハンドルHの1702のフィールドから、この親ディレクトリのiノード番号P1を取り出し、このiノード番号で識別されるディレクトリに存在する名前Fのファイルオブジェクトのiノード番号Iを取得する。次に、1313において親ディレクトリのiノード番号P1とファイルのiノード番号Iからファイルハンドルを構成し、それを要求元クライアント1に返して処理を終了する。

(READ要求の処理)クライアント1は読み出すファイルのファイルハンドルHと、読み出すファイルの親ディレクトリのファイルハンドルHpを引数としてREAD要求を発行する。要求処理手段130がクライアント1からREAD要求を受けると、図8の1321を呼び出し、要求を処理する。まず1322において、ファイルハンドルHのiノード番号フィールド1702からiノード番号Iを取得する。次に、1323において、iノード番号Iにハッシュ関数Hashを適用しファイルグループGを求める。そして、1324で、グループ配置120を参照し、ファイルグループGの格納先ネットワークストレージNを求める。1325で、ネットワークストレージNに対してLOOKUP要求を発行し、READ要求がアクセス対象としたファイルハンドルHで識別されるファイルの実体に対するネットワークストレージNにおけるファイルハンドルHnを取得する。1326において、取得したファイルハンドルHnを引数としてREAD要求をネットワークストレージ装置Nに対して発行する。1327

7において、ネットワークストレージNがREAD要求の返り値である読み出し結果を返すと、得られた読み出し結果を要求元クライアント1に返す。クライアントがWRITE要求を発行した場合もREAD要求の処理とほぼ同じである。1326においてREAD要求を格納先ネットワークストレージNに対して発行する代わりに、WRITE要求を格納先ネットワークストレージNに発行すれば良い。RENAME要求とREMOVE要求も同様である。

(MKDIR要求の処理)クライアント1は生成するディレクトリの名前Dと、生成する親ディレクトリのファイルハンドルHを引数としてMKDIR要求を発行する。要求処理手段130がクライアント1からMKDIR要求を受けると、図9の1331を呼び出し、要求を処理する。まず1332において、ファイルシステム100にファイルハンドルHのiノード番号フィールド1702に格納されているiノード番号によって識別されるディレクトリに、名前Dのディレクトリを生成する。生成したディレクトリのiノード番号をIとする。そして、1333にて、ネットワークストレージのそれぞれにMKDIR要求を発行して名前Dのディレクトリを生成する。そして1334にてファイルハンドルHのiノード番号フィールド1702の値と生成したディレクトリのiノード番号Iとからファイルハンドルを生成し、それを要求元クライアント1に返す。RMDIR要求の処理も同様であり、MKDIR要求をすべてRMDIR要求に変更すれば良い。GETATTR要求とSETATTR要求は、アクセス対象ファイルオブジェクトがディレクトリの場合、MKDIR要求の処理の場合と同様に各ネットワークストレージに対してGETATTR要求またはSETATTR要求を発行してディレクトリ属性の設定を行えば良い。アクセス対象ファイルオブジェクトがファイルの場合、READ要求の処理と同様に、アクセス対象ファイルの格納先ネットワークストレージにGETATTR要求または、SETATTR要求を発行すれば良い。REaddir要求の処理はLOOKUP要求の処理と同様で、ファイルシステム100に対して対象ディレクトリの情報を読み出しそれをクライアントに返せば良い。以上、要求処理手段130の処理の詳細について説明した。ここでは例題として、クライアント1が図3のような構造を持つディレクトリとファイルを生成しそれにアクセスする場合を示す。まず、仮想一元化ネットワークストレージシステム8にディレクトリやファイルがまったく存在しないと仮定する。ファイルグループは0~8までの9つを設ける。また、グループマップとして以下のハッシュ関数

$$\text{Hash}(I) = I/9$$

を用いる。クライアント1はまずマウント要求を仮想一元化装置3に発行し、仮想一元化装置3の要求処理手段130はマウントポイントのファイルハンドルを返す。

次にクライアント1はそのマウントポイントのファイルハンドルと生成するディレクトリの名前d1を引数としてMKDIR要求を発行する。仮想一元化装置3の要求処理手段130はファイルシステム100及び各ネットワークストレージ5、6、7にディレクトリを生成する。次にクライアントは生成したd1のファイルハンドルとディレクトリ名d2を引数にMKDIR要求を発行すると、仮想一元化装置3の要求処理手段130はファイルシステム100と各ネットワークストレージ5、6、7にディレクトリd2を生成する。次にクライアントはディレクトリd2のファイルハンドルとファイル名f1を引数としてCREATE要求を発行する。すると仮想一元化装置3の要求処理手段130はまずファイルシステム100のディレクトリd2において名前f1を持つサイズ0のダミーファイルを生成する。生成したディレクトリやファイルのiノードとディレクトリエントリの構造は図2のようになっているものと仮定する(この図には生成前のディレクトリd3とファイルf2、f3に相当するiノードやディレクトリエントリが含まれている)。ファイルf1に相当するiノードは1005でありそのiノード番号は10番である。そこで要求処理手段130はiノード番号10に前述のハッシュ関数①を適用してファイルf1の属するファイルグループが1番であることが分かる。次にグループ配置120のファイルグループ1番の列を参照すると、そのファイルの格納先ネットワークストレージが1番(図1の5)であることが分かる。従って要求処理手段130はネットワークストレージ5にCREATE要求を発行して、ファイルf1の実体をディレクトリd2の下に生成する。同様にクライアント1がファイルf2のCREATE要求を発行すると、ファイルシステム100において名前f2を持つダミーのファイルが生成する。生成したダミーファイルのiノードが図2の1006でありそのiノード番号が14であるので、要求処理手段130はこれに前述のハッシュ関数①を適用して、このダミーファイルの属するファイルグループが5番であることが分かる。次にグループ配置120を参照して格納先ネットワークストレージが2番(図1の6)であることが分かるので、ネットワークストレージ6にCREATE要求を発行してディレクトリd2の下にファイルf2の実体を生成する。同様にしてディレクトリd3をd1の下に生成し、ディレクトリd3の下にファイルf3を生成すると、ファイルシステム100の構造は図2となり、各ファイルf1、f2、f3の配置は図1の5、6、7のストレージ11に記載の通りとなる。次にクライアント1がファイルf3のREAD要求を発行する場合を考える。クライアント1はまずマウントポイントのファイルハンドルとディレクトリ名d1を引数としてLOOKUP要求を発行し、要求処理手段130はディレクトリd1のファイルハンドルをクライアント1に返す。次にクライアント1は

はディレクトリd1のファイルハンドルとディレクトリ名d3を引数としてLOOKUP要求を発行し、要求処理手段130はディレクトリd3のファイルハンドルをクライアント1に返す。次に、クライアント1はディレクトリd3のファイルハンドルとファイル名f3を指定してLOOKUP要求を発行すると、要求処理手段130はファイルf3のファイルハンドル(ファイルシステム100におけるファイルf1のダミーファイルのiノード番号とその親ディレクトリのiノード番号から構成される)をクライアントに返す。そこでクライアント1はディレクトリd3のファイルハンドルと、ファイルf3のファイルハンドルを引数としてREAD要求を発行する。要求処理手段130はそれに対して、ファイルf3のファイルハンドルからファイルf3に相当するダミーファイルのiノード番号を取り出す。iノード番号は16番であるから、それにハッシュ関数①を提供してファイルグループ7を得る。グループ配置120を参照するとファイルグループ7の格納先ネットワークストレージは3(図1の7)であるから、要求処理手段130はネットワークストレージ3にREAD要求を発行してファイルf3を読み出す。要求処理手段130はネットワークストレージ3から読み出したデータが返ると、それを要求元クライアント1に返す。以上のように、クライアントは単に通常のネットワークストレージへアクセスするように要求を発行すると、仮想一元化装置130がアクセス対象ファイルの格納先ネットワークストレージを求め、そのネットワークストレージにファイルアクセス処理を送ってファイルアクセスを行なう。このためクライアントからは仮想一元化ネットワークストレージシステム8があたかも一つのネットワークストレージとして見える。

(容量監視手段)容量監視手段は定期的に各ネットワークストレージの残容量を求めてそれを図10の容量テーブル1500に書き込むと共に、いずれかのネットワークストレージの残容量が定められた値を下回った場合に、後述の移動手段を起動してファイルの移動を行ったり、あるいは、管理者に対して新しいネットワークストレージの追加を催促するメッセージを送ったりする。容量テーブル1500は各ネットワークストレージの残容量に関する最新の情報を記録するテーブルである。1501の行は各ネットワークストレージの識別子を示す。1502の行は、各ネットワークストレージの最大容量(ファイルの一つも格納しない場合の使用可能容量)を記録する。1503の行は、各ネットワークストレージの現在の残容量を示す。例えば図10の例では、容量の単位をGBと定めると、ネットワークストレージ1、2、3の最大容量は共に20GBであり、残容量はそれぞれ12GB、8GB、14GBである。1503の行は残容量に替えて使用容量を記録しても良い。図11に容量監視手段の容量監視処理のフローチャート(仮想一元化装置130)

3の起動時に1511を実行する新しいスレッドまたはプロセスを生成し、要求処理手段130によるファイルアクセス処理と並列に各ネットワークストレージの容量を監視する。まず1512でフラグFoverを0に初期化する。Foverは後述する移動手段による平滑化処理の開始を指示するフラグである。次に1513で各ネットワークストレージの残容量を取得し、得られた値を容量テーブル1500の1503の行に書き込む。残容量の計測は、各ネットワークストレージにSTATFSプロシージャを発行して取得しても良いし、その他ネットワーク経由で各ネットワークストレージの残容量を取得できればどんな方法を用いても良い。次に、1514において、1513で取得した各ネットワークストレージの残容量が予め定められた閾値Tを下回るか調べ、下回る場合は1515以降を実行する。閾値Tは0以上かつネットワークストレージの最大容量よりも小さな値を指定する。例えば、容量が100GBのネットワークストレージの場合、1GB~10GBが妥当と考えられるが、それ以外の値でも良い。1515では残容量が閾値Tを下回ったネットワークストレージNを移動元ネットワークストレージNsに設定し、1516でFoverフラグに1を設定する。移動手段140はこのフラグを常時監視しており、値が1に設定されると後述の平滑化処理を起動し、移動元に設定されたネットワークストレージNのファイルグループの幾つかを他のネットワークストレージに移動し各ネットワークストレージ間の残容量の平滑化を計る。その後1517において一定時間スリープし、1513の処理へ戻る。一方、1514において全ての残容量が閾値Tを下回らなかった場合は、1517で一定時間スリープし、1513の処理へ戻る。1517でスリープする時間は、1秒程度を最小とし、1分出会っても良いし、1時間であっても良いし、1日であっても良いし、その他なんでも良い。スリープ時間が短いほど、容量の急激な増加に対して正確に状況を把握できるようになるが、逆に容量の監視処理自身が頻繁に動くため、仮想一元化装置の処理が重くなる。従って、容量の増減の度合いによって、適切なスリープ時間を設定する。なお、1514の条件は、ネットワークストレージの最大残容量と最小残容量の差が予め定められた閾値Tdより大きいかどうかという条件にしても良い。このとき、閾値Tdは0からネットワークストレージの容量までの値を指定する。例えば、ネットワークストレージの容量が100GBであれば、1GB~10GBが妥当な値であるが、それ以外でも良い。上記のように1514の条件を変更した場合は、このとき1515の処理を残容量の最も小さなネットワークストレージを移動元ネットワークストレージに設定するよう変更する。さらに、1514の条件を全てのネットワークストレージの総残容量が予め定められた閾値Ttを下回るか

16で管理者に対して残容量が残り少なく新たにネットワークストレージを追加することを催促するメッセージを送るように変更しても良い。メッセージを送る手段は、sendmailのような公知の技術を使用して管理者にメールを送信しても良いし、管理手段180を介して管理者のインタフェースにメッセージを出しても良い。

(移動手段)移動手段140は、ネットワークストレージを追加したり削除して、仮想一元化ネットワークストレージシステム8の構成を変更する場合に、構成変更の対象となるネットワークストレージに格納されているファイルの移動を行なう。これによって、仮想一元化ネットワークストレージシステムの構成変更が柔軟に行なえる。また、移動手段140は各ネットワークストレージの残容量に規定以上のばらつきが生じた場合に、ファイルをネットワークストレージ間で移動し、残容量の平滑化を図る。ファイルを移動しないと、いずれかのネットワークストレージが一杯になった段階で、他のネットワークストレージに空きがあっても、一切ファイルの書き込みができなくなる。これに対して、上記のようにファイルを移動することによって、ほぼ全てのネットワークストレージの容量を使い切るまで使用できるようになる。移動手段140には、基本移動機能、ネットストレージ追加時の移動機能、ネットストレージ削除時の移動機能、平滑化機能の四つの機能がある。各機能に対応して、それぞれ移動処理、追加処理、削除処理、平滑化処理がある。それぞれの処理の詳細を以下に示す。

(移動処理)移動処理は、ファイルをファイルグループ単位で現在格納されているネットワークストレージから別のネットワークストレージに移動する。図12に移動処理のフローを示す。移動対象ファイルグループGmと、移動先ネットワークストレージNdを設定して移動処理1401を呼び出すと、移動手段140はファイルグループGmを現在格納されているネットワークストレージから、移動先ネットワークストレージNdに移動する。移動処理はまず1402でグループ配置120を参照し、移動対象グループGmのファイルが現在格納されているネットワークストレージNsを求める。次に、ファイルシステム100のファイルをルートディレクトリ／から一つずつ探索し、ファイルグループGmに属するファイルを特定する。このためまず1403でファイルシステム100に探索すべきファイルFが存在するか調べ、存在する場合は1404でファイルFのiノード番号Iを求める。1405でファイルFのiノード番号Iにハッシュ関数Hashを適用してファイルFの属するファイルグループを求め、それが移動対象ファイルグループGmと一致する場合は、1406でファイルFを移動元ネットワークストレージ装置Nsから移動先ネットワークストレージNdに移動する。移動は、移動手段140が、移動元ネットワークストレージNsにファイルFのiノード番号Iと要求を発行してファイルFを特定し、移動先ネットワークストレージNdに発行してファイルFを書き込み、最後に移動元ネットワークストレージのファイルFを削除する。あるいは、FTP(File Transfer Protocol)を利用してファイルを移動元から移動先にコピーした後、移動元のファイルを削除しても良い。その他、ファイルを移動元ネットワークストレージから移動先ネットワークストレージに移動できればどんな方法を用いても良い。移動が終わると1403に戻って処理を続ける。1403で探索すべきファイルが存在しない場合は、移動対象ファイルグループGmに属する全てのファイルを移動し終えているので、1407でグループ配置120のファイルグループGmの欄に移動先ネットワークストレージNdを書き込み、処理を終了する。本発明の仮想一元化ネットワークストレージシステム8の管理者がユーザインタフェース経由で管理手段180に対し、移動対象ファイルグループGmと移動先ネットワークストレージNdを指定し移動の指示を与えたときに呼び出されるか、あるいは、追加処理、削除処理、平滑化処理の処理過程において呼び出される。移動処理後に、移動処理によって格納先ネットワークストレージが変更されたファイルFに対し、クライアント1からファイルアクセス要求があった場合、要求処理手段130は移動手段140が1407で変更した後のグループ配置120を参照して、格納先ネットワークストレージを決定する。このため、ファイルFの格納先ネットワークストレージが変更されても、クライアント1は正しくファイルFにアクセスできる。なお、移動手段140がファイルグループGmを移動している最中に、クライアント1がファイルグループGmに属するファイルへアクセス要求を発行した場合、要求処理手段130は、ファイルグループGmの移動が終了するまで、クライアントからのファイルアクセス要求を実行しない。

(追加処理)本発明の仮想一元化装置3は仮想一元化ネットワークストレージシステム8の容量が不足したとき、新しくネットワークストレージ装置を追加することで、容量を増大することができる。仮想一元化ネットワークストレージシステム8の管理者は、追加するネットワークストレージをネットワーク4に接続し、ユーザインタフェースを介して管理手段180にアクセスし、追加処理の開始を指示する。すると管理手段180は、移動手段140の追加処理(図13の1411)を呼び出す。追加処理は、仮想一元化ネットワークストレージシステム8にネットワークストレージを追加した場合に、既存のネットワークストレージを格納先とするファイルグループの幾つかを追加したネットワークストレージに移動し、これによって追加したネットワークストレージを仮想一元化ネットワークストレージシステムに取り込む。追加処理は、移動手段140が、移動元ネットワークストレージNsにファイルFのiノード番号Iと要求を発行してファイルFを特定し、移動先ネットワークストレージNdに発行してファイルFを書き込み、最後に移動元ネットワークストレージのファイルFを削除する。あるいは、FTP(File Transfer Protocol)を利用してファイルを移動元から移動先にコピーした後、移動元のファイルを削除しても良い。その他、ファイルを移動元ネットワークストレージから移動先ネットワークストレージに移動できればどんな方法を用いても良い。移動が終わると1403に戻って処理を続ける。1403で探索すべきファイルが存在しない場合は、移動対象ファイルグループGmに属する全てのファイルを移動し終えているので、1407でグループ配置120のファイルグループGmの欄に移動先ネットワークストレージNdを書き込み、処理を終了する。本発明の仮想一元化ネットワークストレージシステム8の管理者がユーザインタフェース経由で管理手段180に対し、移動対象ファイルグループGmと移動先ネットワークストレージNdを指定し移動の指示を与えたときに呼び出されるか、あるいは、追加処理、削除処理、平滑化処理の処理過程において呼び出される。移動処理後に、移動処理によって格納先ネットワークストレージが変更されたファイルFに対し、クライアント1からファイルアクセス要求があった場合、要求処理手段130は移動手段140が1407で変更した後のグループ配置120を参照して、格納先ネットワークストレージを決定する。このため、ファイルFの格納先ネットワークストレージが変更されても、クライアント1は正しくファイルFにアクセスできる。なお、移動手段140がファイルグループGmを移動している最中に、クライアント1がファイルグループGmに属するファイルへアクセス要求を発行した場合、要求処理手段130は、ファイルグループGmの移動が終了するまで、クライアントからのファイルアクセス要求を実行しない。

(追加処理)本発明の仮想一元化装置3は仮想一元化ネットワークストレージシステム8の容量が不足したとき、新しくネットワークストレージ装置を追加することで、容量を増大することができる。仮想一元化ネットワークストレージシステム8の管理者は、追加するネットワークストレージをネットワーク4に接続し、ユーザインタフェースを介して管理手段180にアクセスし、追加処理の開始を指示する。すると管理手段180は、移動手段140の追加処理(図13の1411)を呼び出す。追加処理は、仮想一元化ネットワークストレージシステム8にネットワークストレージを追加した場合に、既存のネットワークストレージを格納先とするファイルグループの幾つかを追加したネットワークストレージに移動し、これによって追加したネットワークストレージを仮想一元化ネットワークストレージシステムに取り込む。追加処理は、移動手段140が、移動元ネットワークストレージNsにファイルFのiノード番号Iと要求を発行してファイルFを特定し、移動先ネットワークストレージNdに発行してファイルFを書き込み、最後に移動元ネットワークストレージのファイルFを削除する。あるいは、FTP(File Transfer Protocol)を利用してファイルを移動元から移動先にコピーした後、移動元のファイルを削除しても良い。その他、ファイルを移動元ネットワークストレージから移動先ネットワークストレージに移動できればどんな方法を用いても良い。移動が終わると1403に戻って処理を続ける。1403で探索すべきファイルが存在しない場合は、移動対象ファイルグループGmに属する全てのファイルを移動し終えているので、1407でグループ配置120のファイルグループGmの欄に移動先ネットワークストレージNdを書き込み、処理を終了する。本発明の仮想一元化ネットワークストレージシステム8の管理者がユーザインタフェース経由で管理手段180に対し、移動対象ファイルグループGmと移動先ネットワークストレージNdを指定し移動の指示を与えたときに呼び出されるか、あるいは、追加処理、削除処理、平滑化処理の処理過程において呼び出される。移動処理後に、移動処理によって格納先ネットワークストレージが変更されたファイルFに対し、クライアント1からファイルアクセス要求があった場合、要求処理手段130は移動手段140が1407で変更した後のグループ配置120を参照して、格納先ネットワークストレージを決定する。このため、ファイルFの格納先ネットワークストレージが変更されても、クライアント1は正しくファイルFにアクセスできる。なお、移動手段140がファイルグループGmを移動している最中に、クライアント1がファイルグループGmに属するファイルへアクセス要求を発行した場合、要求処理手段130は、ファイルグループGmの移動が終了するまで、クライアントからのファイルアクセス要求を実行しない。

加するネットワークストレージを移動先ネットワークストレージN dに設定する。次に1413において、容量テーブル1500の1503の行を参照し、ストレージの残容量が最も小さなネットワークストレージを移動元ネットワークストレージNsとする。そして1414で、グループ配置120を参照し、移動元ネットワークストレージNsを格納先とするファイルグループを求め、その中からランダムに一つファイルグループを選択し、それを移動対象ファイルグループGmに設定する。そして、前述の図12の移動処理1401を呼び出し、移動対象ファイルグループGmを移動元ネットワークストレージNsから追加したネットワークストレージである移動先ネットワークストレージNdに移動する。移動が終了すると、1416において、容量テーブル1500を参照し、移動先ネットワークストレージNdの残容量ともっとも少ない残容量を求め、その差が予め定められた閾値Tcを超えれば1413に戻って処理を続け、そうでなければ処理を終了する。閾値Tcは、100MB、500MB、あるいは、1GBなど、なんでも良い。ネットワークストレージ追加時の移動処理にかかる時間を短縮するには、閾値に大きな値を設定する。時間がかかっても各ネットワークストレージ間の残容量の格差を小さくしたい場合は閾値に小さな値を設定する。なお、1413において移動元ネットワークストレージを選択する方法は、ランダムに選択しても良いし、予め定められた順に選択しても良いし、どんな方法を用いても良い。また、1414においてファイルグループGmを選択する方法は、予め定められた順に選択しても良いし、その他どんな方法を用いても良い。また、1416において残容量により終了判定するのではなく、移動したファイルグループの数が予め定めておいた値を超えたら終了するという条件にしても良い。追加処理の終了後、クライアント1がファイルアクセス要求を発行し、そのアクセス要求の対象が、上記追加処理によって追加ネットワークストレージに移動したファイルグループに属するファイルである場合、要求処理手段130はクライアント1からのアクセス要求を追加しネットワークストレージに送り、追加ストレージにてファイルアクセス処理を行う。このように、クライアント1から見ると、仮想一元化ネットワークストレージシステムにネットワークストレージを追加しても、ファイルアクセスの手順は追加前と変わらない。クライアント1から見た場合の唯一の違いは、仮想一元化ネットワークストレージシステムの容量が追加したネットワークストレージの容量分だけ増えたように見えることである。

(削除処理)本実施例の仮想一元化装置3は仮想一元化ネットワークストレージシステム8の容量があまっていて利用しないなどの理由で、システム内のネットワークストレージを取り除き、容量を小さくすることができる。

るのは、システム全体の空き容量が削除するネットワークストレージの容量より多く存在する場合に限られる。仮想一元化ネットワークストレージシステム8の管理者は、ユーザインタフェースを介して管理手段180にアクセスし、削除するネットワークストレージを指定して削除処理の開始を指示する。すると管理手段180は、移動手段140の削除処理(図14の1421)を呼び出す。削除処理は、仮想一元化ネットワークストレージシステム8内のネットワークストレージ5、6、7のいずれかを削除する場合に、削除するネットワークストレージを格納先とする全てのファイルグループをその他のネットワークストレージに移動し、削除するネットワークストレージを仮想一元化ネットワークストレージシステムから論理的に切り離す。処理の詳細フローを図14に示す。まず、1422で削除するネットワークストレージを移動元ネットワークストレージNsに設定する。次に1423で、容量テーブル1500の1503の行を参照し、ストレージの残容量が最も大きなネットワークストレージを移動先ネットワークストレージNdとする。そして1424で、グループ配置120を参照し、移動元ネットワークストレージNsを格納先とするファイルグループを求め、その中からランダムに一つファイルグループを選択し、それを移動対象ファイルグループGmに設定する。そして、前述の図12の移動処理1401を呼び出し、移動対象ファイルグループGmを削除するネットワークストレージである移動元ネットワークストレージNsから移動先ネットワークストレージNdに移動する。移動が終了すると、1426で、移動元ネットワークストレージNsにファイルグループが存在すれば1423に戻って処理を続け、そうでなければ処理を終了する。1423の移動先ネットワークストレージの選択方法は、ネットワークストレージの残容量が移動するファイルグループの容量よりも大きければ、ランダムに選択しても良いし、予め順番を決めてその順に選択しても良い。また、1424においてファイルグループGmの選択方法は、予め定められた順番に選択しても良いし、その他どんな方法で選択しても良い。上記削除処理が終了すると、管理手段180はインタフェースを介して管理者に削除の終了を報告する。この報告を受けて管理者は論理的に切り離されたネットワークストレージをネットワーク4から物理的に切り離す。削除処理の終了後、クライアント1がファイルアクセス要求を発行し、そのアクセス要求の対象が、削除前に削除したネットワークストレージを格納先とするファイルグループに属するファイルであった場合、要求処理手段130はクライアント1からのアクセス要求を削除したネットワークストレージではなく、移動先ネットワークストレージに送り、ファイルアクセス処理を行う。このように、クライアント1から見ると、仮想一元化ネットワークスト

も、ファイルアクセスの手順は削除前と変わらない。クライアント1から見た場合の唯一の違いは、仮想一元化ネットワークストレージシステムの容量が削除したネットワークストレージの容量分だけ減ったように見えることである。

(平滑化処理)クライアント1が仮想一元化ネットワークストレージシステム8に対してファイルの書き込み要求を発行し、書き込み対象ファイルの格納先ネットワークストレージに空きがない(残容量が0)場合、他のネットワークストレージに大量の空きがあっても、そのファイルの書き込みはできない。このような状況に対処するため、仮想一元化装置3は、ネットワークストレージの残容量が少なくなったり、ネットワークストレージ間の残容量の格差が広がると、平滑化処理を起動して、各ネットワークストレージの残容量が均等になるようにファイルグループの移動を行う。平滑化処理は、仮想一元化装置3の起動時に移動手段140が独立したスレッドまたはプロセスとして平滑化処理(図15の1431)を起動する。平滑化処理の詳細を図15に示す。平滑化処理はバックグラウンドで動いているが、1432でF o v e rというフラグが0である間は処理を行わない。このフラグは、前述の(容量監視手段)のところで述べたように、ネットワークストレージの残容量が定められた閾値を下回った場合、そのネットワークストレージを移動元ネットワークストレージNsとしF o v e rを1に設定する。あるいは、ネットワークストレージの最大残容量と最小算容量の差が閾値を超えた場合に、残容量の最も小さなネットワークストレージを移動元ネットワークストレージに設定し、F o v e rを1に設定する。F o v e rが1に設定されると、平滑化処理は1433で移動元ネットワークストレージNsを格納先とするファイルグループのうちの一つをランダムに選択して移動対象ファイルグループGmとし、1434で残容量が最大のネットワークストレージを移動先ネットワークストレージNdとし、1435で1401の移動処理を呼び出しファイルグループGmを移動元ネットワークストレージNsから移動先ネットワークストレージNdに移動する。1436で、ネットワークストレージの最大残容量と最小残容量の差が前述の追加処理で説明した閾値Tcの値より大きければ、1433に戻って処理を続け、小さくなれば1437でF o v e rフラグを0に設定して、1432に戻る。なお、1433においてNsからGmを選択する方法は、予め定められた順番に選択する方法であっても良いし、その他どのような方法であっても良い。また、1434において、Ndに設定するネットワークストレージは、移動元ネットワークストレージNs以外のネットワークストレージを選択できれば、どのような方法を用いても良い。上記の処理によって、ネットワークストレージ間の残容量の差が小さくなり、特定の

書き込みができなくなるという問題を解決することができる。以上自動的に平滑化処理を起動する場合について説明したが、管理者がユーザインタフェースを介して管理手段180にアクセスし、平滑化処理の実行を指示しても良い。このとき、管理者は移動元ネットワークストレージを引数として与える。移動手段140は図15の1433を呼び出して平滑化の処理を行い。1436で条件が満足されなくなると処理を終了する。なお、管理手段180はユーザインタフェースを介して、容量テーブル1500の情報を管理者にリアルタイムに提供し、管理者が平滑化処理や追加処理や削除処理を行うべきかどうか判断する手助けにすることが望ましい。

(統合手段)本発明の仮想一元化ネットワークストレージシステム8でない、一般的なネットワークストレージを用いているサイトにおいて、ネットワークストレージの容量が不足すると、新規に既存ネットワークストレージを購入するのが一般的である。しかしこの場合、クライアントからは、既設ネットワークストレージと追加ネットワークストレージが独立して見えるため、使い勝手が悪い。このような場合に対処するため、本発明の仮想一元化手段3は統合化手段180によって、既設ネットワークストレージを取り込んで仮想一元化ネットワークストレージシステムを構成し、既設ネットワークストレージのファイルを含みかつより容量の大きな仮想一元化ネットワークストレージを提供する。図1のネットワーク2に既設ネットワークストレージが接続されており、クライアント1のファイルが既設ネットワークに格納されているものとする。管理者はネットワーク2に仮想一元化ネットワークシステム8を接続し、ユーザインタフェースを介して仮想一元化装置3の管理手段180にアクセスし、統合する既設ネットワークストレージを指定して統合の開始を指示する。すると、仮想一元化装置3の統合手段160は図16の1601を呼び出し、統合処理を開始する。統合処理はまず1602でグループ配置120の各ファイルグループに対応したネットワークストレージの欄に、既設ネットワークストレージ又は追加したネットワークストレージの中の一つをそれぞれ書き込む。例えば、ファイルグループが1~4の4つあり、ネットワークストレージが既設と追加の2つである場合、ファイルグループ1と2に既設ネットワークストレージを、ファイルグループ3と4に追加ネットワークストレージをそれぞれ対応付ける。あるいは、全てのファイルグループに既設ネットワークストレージを割り当てても良いし、その他どのような割り当てでも良い。このグループ配置120へのネットワークストレージの書き込み方によって、既設ネットワークストレージを仮想一元化ネットワークストレージシステム8に統合し運用を開始した直後における、ファイルグループの格納先ネットワークストレージが決定される。次に1603で、図

既設ネットワークストレージに未探索のファイルオブジェクトFが存在するか調べ、存在する場合は1613でファイルオブジェクトFがディレクトリかどうか調べ、ディレクトリなら1614で仮想一元化装置3のファイルシステム100にディレクトリFを生成して1612に戻って処理を継続する。1613で一方ファイルオブジェクトFがディレクトリでなくファイルである場合は、1615でファイルFと同じ名前と属性を持ちサイズ0のダミーファイルをファイルシステム100に生成し、1612に戻って処理を継続する。1612で未探索のファイルオブジェクトが存在しなければ、図16の1603の処理は終了する。次に、1604において図18の1621を呼び出す。すると1622でファイルシステム100に未探索のファイルFが存在するかどうかが調べ、存在する場合は1623でファイルFのiノード番号Iを求める。1624でファイルFの属するファイルグループGを、iノード番号Iにグループマップ110で与えられたハッシュ関数Hashを適用して求める。1625で、グループ配置120を参照しファイルグループGの格納先ネットワークストレージNを求める。この格納先ネットワークストレージは図16の1602で設定したものである。統合中の現時点において、全てのファイルグループは既設ネットワークストレージに格納されているから、1625で求めたファイルグループGの格納先ネットワークストレージNが既設ネットワークストレージである場合には(1626)グループ配置の設定と現状とが一致するので何もしなくて良く、1622に戻って処理を続ける。もし、1626でNが既設ネットワークストレージでない場合には、1627において、ファイルFを既設ネットワークストレージからネットワークストレージNに移動する。移動の方法は、既設ネットワークストレージにREAD要求を発行してファイルFを読み出し、ネットワークストレージNに読み出したデータを引数としてWRITE要求を発行してファイルFを書き込み、最後に既設ネットワークストレージのファイルFを削除すればよい。移動が終了すると1622に戻って処理を続ける。1622において、未探索のファイルが存在しなくなると図16の1604の処理が終了し、統合処理全体も終了する。なお、統合手段160による統合処理の実行中は、要求処理手段によるファイルアクセス処理や移動手段による追加・削除・平滑化処理を停止する。統合処理が終了したあと、クライアント1が既設ネットワークストレージの代わりに仮想一元化装置3に対してファイルアクセスコマンドを発行すると、仮想一元化装置3はアクセス対象ファイルFの格納先ネットワークストレージに対しアクセス要求を発行してファイルアクセスを行うため、クライアントにとっては既設ネットワークストレージにアクセスするのと、既設ネットワークストレージを統合化した

する場合とで違いは無く、ただ後者の方が容量が多いように見える。以上、仮想一元化装置3の各構成要素について説明した。これらの各構成要素を搭載した仮想一元化装置の実装例を図19に示す。CPU20、コントローラ21、メモリ22、I/Oインタフェース23、ネットワークインタフェース24、ディスク装置25から構成されるサーバ、ワークステーション、あるいはPCをハードウェアとして、前述のUNIX系オペレーティングシステム、例えばLinuxを搭載し、ファイルシステム100、グループマップ110、グループ配置120はそれぞれデータとしてディスク装置25に格納し、要求処理手段130、移動手段140、容量監視手段150、統合手段160、管理手段180はそれぞれプログラムコードとしてメモリ22に配置し、CPU20がそれらを実行することで仮想一元化装置として機能する。

<実施例2>図20は本発明の第2の実施例を含む情報システムの全体構成を示す図である。本実施例は図1の実施例1の変形であるから、実施例1との相違部分についてのみ説明する。14は図1の仮想一元化ネットワークストレージシステム8と同等の機能を持つ仮想一元化ネットワークストレージシステムである。14のネットワーク4と一つ以上のネットワークストレージ6、7は図1の8の仮想一元化ネットワークストレージシステムと同じであり、異なるのは12の仮想一元化装置である。本実施例の仮想一元化装置12は、ファイルシステム100、グループマップ110、グループ配置120、要求処理手段130、移動手段140、容量監視手段150、統合手段160、管理手段180は実施例1の仮想一元化装置3と同一であり、ネットワークストレージ5、6、7の同一のリモート制御手段10とファイルシステムを構築したストレージ装置11を持ち、ネットワークストレージとしても機能するネットワークストレージ機能付き仮想一元化装置である点が異なる。8の仮想一元化装置はそれ単独では仮想一元化ネットワークストレージシステムを構成することはできず、他に必ずネットワークストレージを接続する必要があるが、本実施例のネットワークストレージ機能付き仮想一元化装置の場合は、それ単体で仮想一元化ネットワークストレージシステムを構成することが可能であり、より安価に仮想一元化ネットワークストレージシステムを提供できる。一方、実施例2はネットワークストレージを含むため、仮想一元化装置12が管理データの他にファイルの実体のアクセス処理も行う必要があり、したがってCPU20の負荷が高くなり易い。すなわち、実施例1の方がより高い性能を出すことができるという点で優れている。本実施例の場合グループ配置120にはネットワークストレージ装置としても機能する仮想一元化装置12も特定のファイルグループの格納先として登録する。ク

求処理手段１３０がそれを受け、アクセス対象ファイルの格納先ネットワークストレージを求め、求めた格納先ネットワークストレージ装置にファイルアクセス要求を発行してファイルアクセスを行う一連の処理は実施例１と同じである。ただし本実施例の仮想一元化装置はネットワークストレージとしても機能し、格納先ネットワークストレージが一元管理装置１２であった場合、要求手段１３０は自身のリモート制御手段１０に対してファイルアクセス要求を発行し、リモートアクセス手段１０は要求処理手段１３０からの要求を受けてストレージ装置１１に対してファイルアクセス処理を行う。なお、本実施例２のネットワークストレージ機能付き仮想一元化装置１２のストレージ装置１１を２つのパーティションに分割し、その一方にファイルシステム１００を搭載して管理用のファイルシステムとして用い、もう一方のパーティションにもファイルシステムを搭載してファイルの実体を格納するために用いても良い。

<実施例３>図２は本発明の第３の実施例を含む情報システムの全体構成を示す図である。本実施例は図１の実施例１の変形であるから、実施例１との相違部分についてのみ説明する。まず図１では仮想一元化装置３の内部に存在した要求処理手段１３０に少し手を加えて、仮想一元化装置１３と分離する。この要求処理手段１３１は、サーバ、ワークステーション、ＰＣ等の公知のハードウェア上にLinux等の既存オペレーティングシステムを搭載し、そこに要求処理手段１３１を搭載することで実装する。１３１はクライアント１とＮＦＳプロトコルを用いてファイルアクセス処理を行う。クライアント１は複数ある要求処理手段１３１の何れかにファイルアクセス要求を発行する。要求処理手段１３１は自身の中にファイルシステム１００等の管理情報を持たず、クライアント１がファイルアクセス要求を発行すると、ネットワーク２を介して一元管理装置１３の要求処理手段１３２にアクセスしてアクセス対象ファイルの格納先ネットワークストレージを求める。そして求めた格納先ネットワークストレージにファイルアクセス要求を発行してファイルアクセスを行う。ファイルアクセス要求が終了すると要求処理装置１３１はネットワークストレージから得られたファイルアクセス要求の結果を要求元クライアント１に返す。ＮＦＳの各プロシージャの処理に関しては、自身のファイルシステム、グループマップ、及び、グループ配置にアクセスする代わりに、上記のようにネットワーク経由で仮想一元化装置３の要求処理手段１３２にアクセスし、ファイルシステム１００、グループマップ１１０、グループ配置１２０にアクセスして所望のデータを取得すること意外実施例１で説明した処理と同様である。一元管理装置１３の要求処理手段１３２は要求処理手段１３１からの管理情報へのアクセス要求を受け、ファイルシステム１００、グループマップ１１

他、仮想一元化装置１３における各種の手段の処理は実施例１と同じである。本実施例３はクライアント１からファイルアクセス要求を受ける要求処理手段１３１を、仮想一元化装置１３と独立させ、かつ、複数設けることによって、実施例１と比べてより多くのクライアントからのリクエストに答えられるようになる。すなわち高い性能が要求される環境での使用は本実施例の形態をとると良い。なお、要求処理手段を複数持たせると、クライアント１から本発明の仮想一元化ネットワークストレージシステムに複数のエントリポイントがあるように見える。そこで、複数の要求手段１３１の前段に負荷分散装置等の公知な技術を導入してエントリポイントを一元化しても良い。

【０００８】

【発明の効果】本発明のネットワークストレージシステムの仮想一元化方法及び装置はファイルの格納先ネットワークストレージをファイルグループを単位として管理する。ファイルからそのファイルが属するファイルグループを求めるファイルマップを固定的に与えておいて、ファイルグループとその格納先ネットワークストレージとの対応をグループ配置として管理し、ネットワークストレージの追加や削除等の構成変更があった場合には、このグループマップを変更する。このようなファイルグループを単位とした管理方法により、ファイル毎に格納先ネットワークストレージの情報を持つ従来の方法と比べて管理情報が少なくなり、ネットワークストレージの容量を有効に利用できるようになる。また、仮想一元化ネットワークストレージシステムの構成変更時には、ファイルグループを単位としてファイルの格納先ネットワークストレージを移動する。この際に、グループ配置の移動するファイルグループの格納先ネットワークストレージを変更するだけで良いので、ファイル毎に格納先情報を管理する場合に比べてオーバーヘッドを小さくできる。

【図面の簡単な説明】

【図１】本発明のネットワークストレージ装置の仮想一元化方法の構成を示す図である。

【図２】ファイルシステムの構造を示す図である。

【図３】ファイルシステムの構造を説明するために使用するディレクトリ構造の例題を示す図である。

【図４】グループ配置の構成を示す図である。

【図５】CREATE要求に対する処理の流れを示すフローチャートである。

【図６】ファイルハンドラの構成を示す図である。

【図７】LOOKUP要求に対する処理の流れを示すフローチャートである。

【図８】READ要求に対する処理の流れを示すフローチャートである。

【図９】MKDIR要求に対する処理の流れを示すフローチャートである。

【図10】容量テーブルの構成を示す図である。

【図11】容量の監視処理の流れを示すフローチャートである。

【図12】ファイルグループをネットワークストレージ間で移動する処理の流れを示すフローチャートである。

【図13】ネットワークストレージ装置を追加する場合の処理の流れを示すフローチャートである。

【図14】ネットワークストレージ装置を削除する場合の処理の流れを示すフローチャートである。

【図15】ネットワークストレージ間で容量に不均衡が生じた場合に行う各ネットワークストレージの残容量の平滑化処理のフローチャートである。

【図16】既設ネットワークストレージ装置の仮想一元化ネットワークストレージシステムへの統合方法の処理の流れを説明するフローチャートである。

【図17】既設ネットワークストレージ装置の統合に際して行うファイル識別子の生成方法の処理の流れを示すフローチャートである。

【図18】既設ネットワークストレージ装置の統合に際して行うファイルの移動処理の流れを示すフローチャートである。

【図19】本発明の仮想一元化方法を搭載した計算機システムの図である。

【図20】ネットワークストレージとしての機能を備えた一元管理装置の構成を示す図である。

【図21】複数の要求処理手段を備えた仮想一元化ネットワークストレージシステムを示す図である。

【符号の説明】

- 1 クライアント
2、4 ネットワーク

3 仮想一元化装置

5、6、7 ネットワークストレージ

8、14、15 仮想一元化ネットワークストレージシステム

10 リモート

11 ストレージ装置

12 ネットワークストレージ機能付き仮想一元化装置

13 要求処理部を持たない仮想一元化装置

20 CPU

21 コントローラ

22 主記憶

23 I/Oインタフェース

24 ネットワークインタフェース

25 ストレージ装置

100 ファイルシステム

110 グループマップ

120 グループ配置

130、131、132 要求処理手段

140 移動手段

150 容量監視手段

160 統合手段

170 ファイルハンドル

180 管理手段

1001、1002、1003、1004、1005、

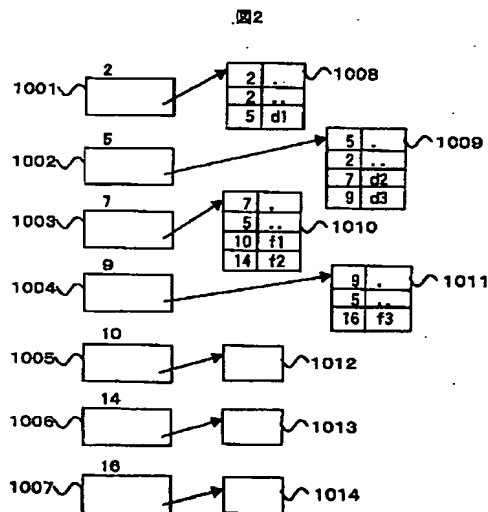
1006、1007 iノード

1008、1009、1010、1011 ディレクトリエントリ

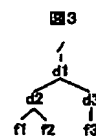
1012、1013、1014 データブロック

1500 容量テーブル。

【図2】



【図3】



【図10】

図10: 容量テーブル

1501	1	2	3	4	5	...
1502	20	20	20	-	-	...
1503	12	8	14	-	-	...

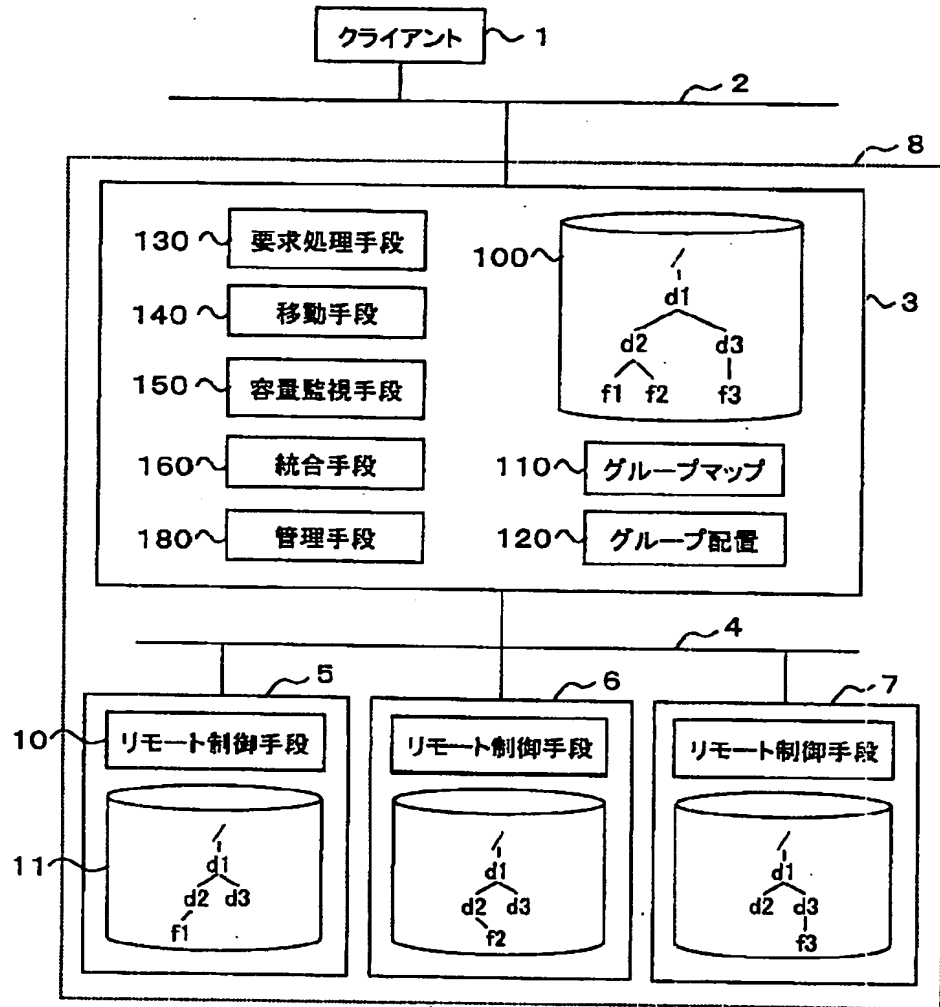
【図4】

図4: グループ配置

1201	0	1	2	3	4	5	6	7	8
1202	1	1	1	2	2	2	3	3	3

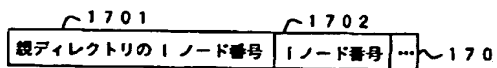
【図1】

図1



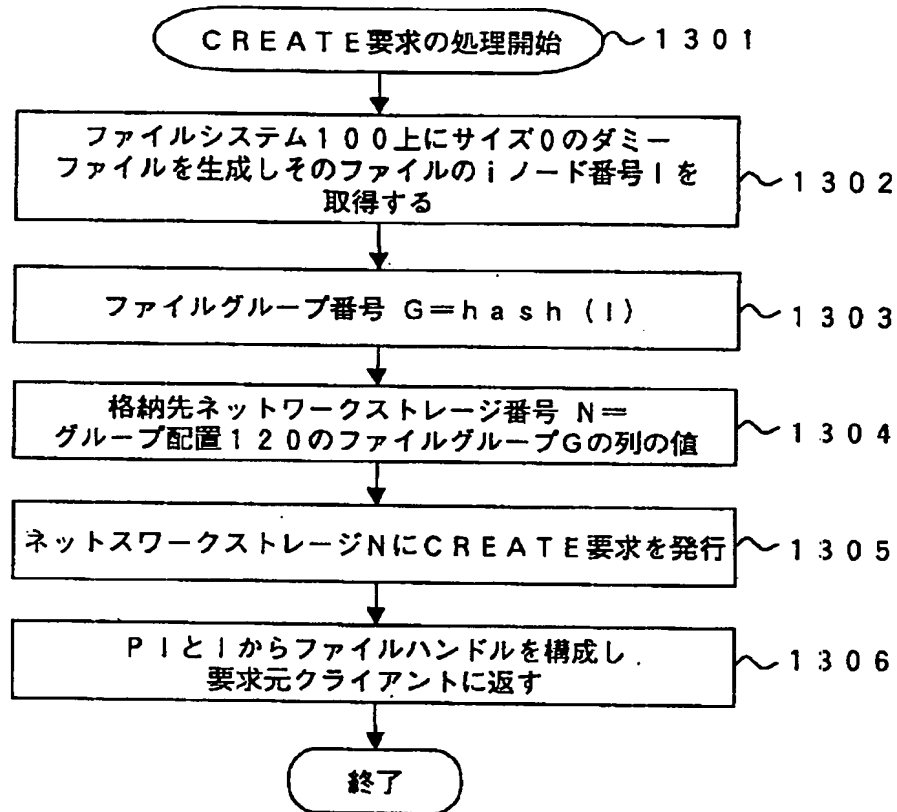
【図6】

図6



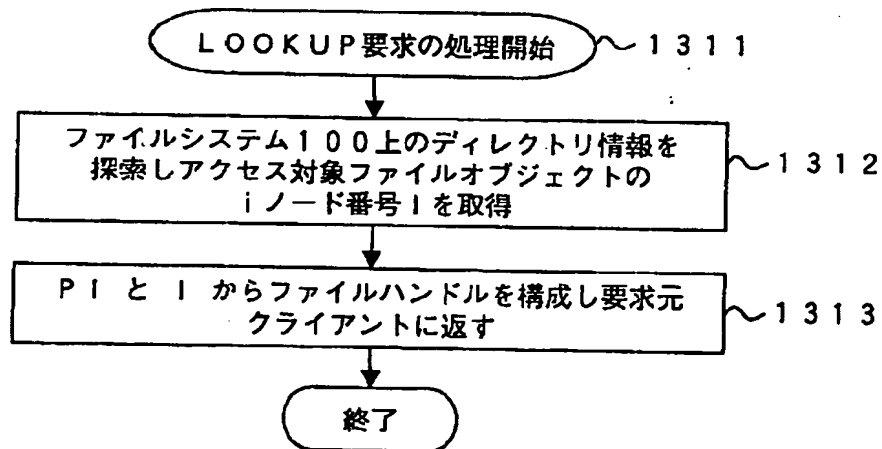
【図5】

図5



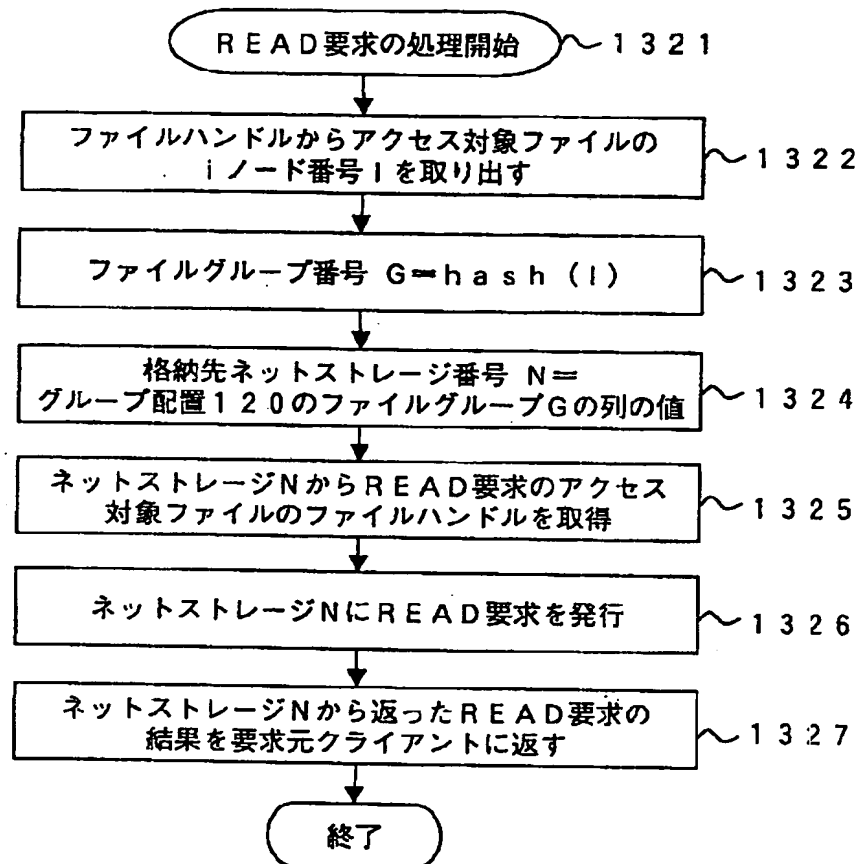
【図7】

図7



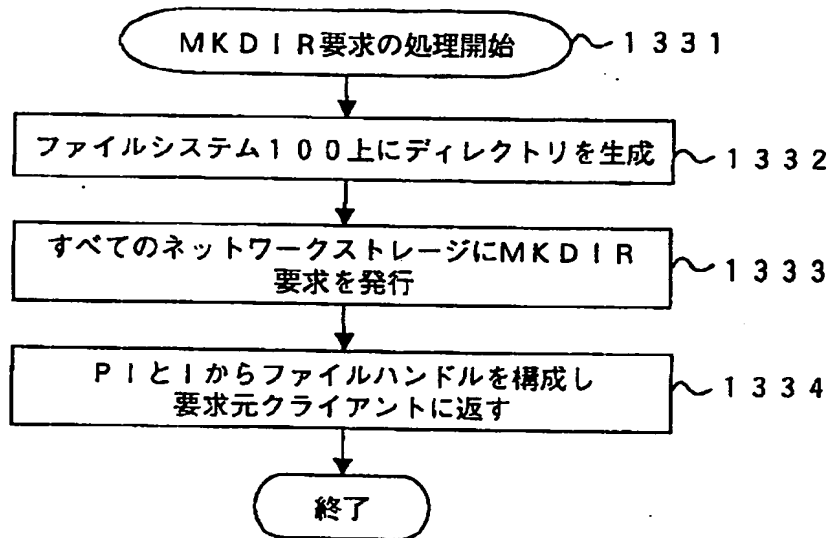
【図8】

図8



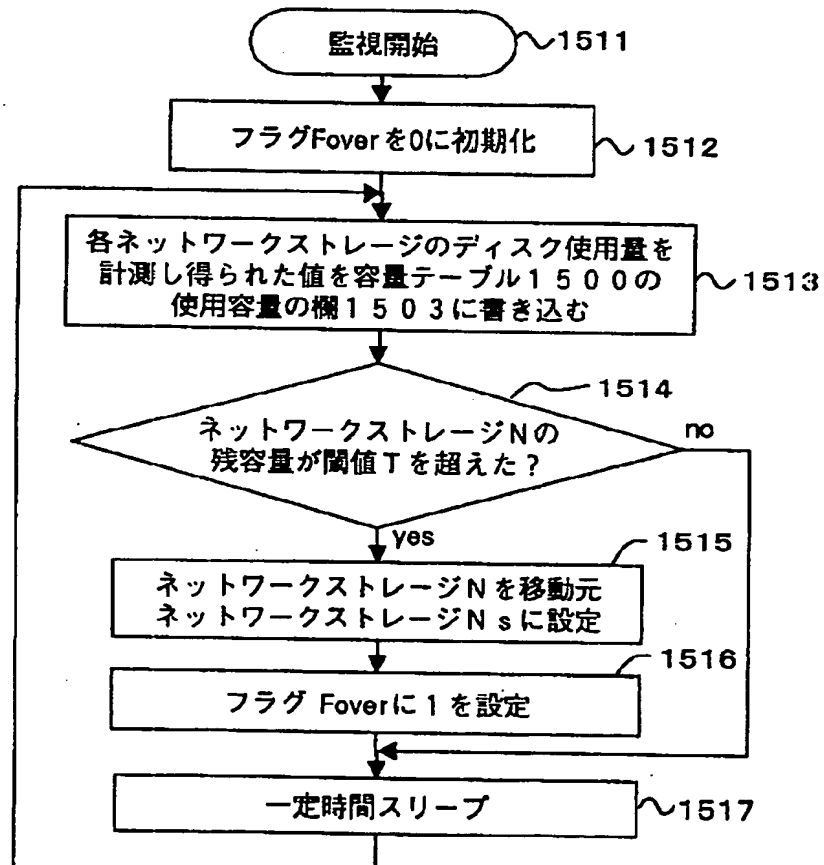
【図9】

図9



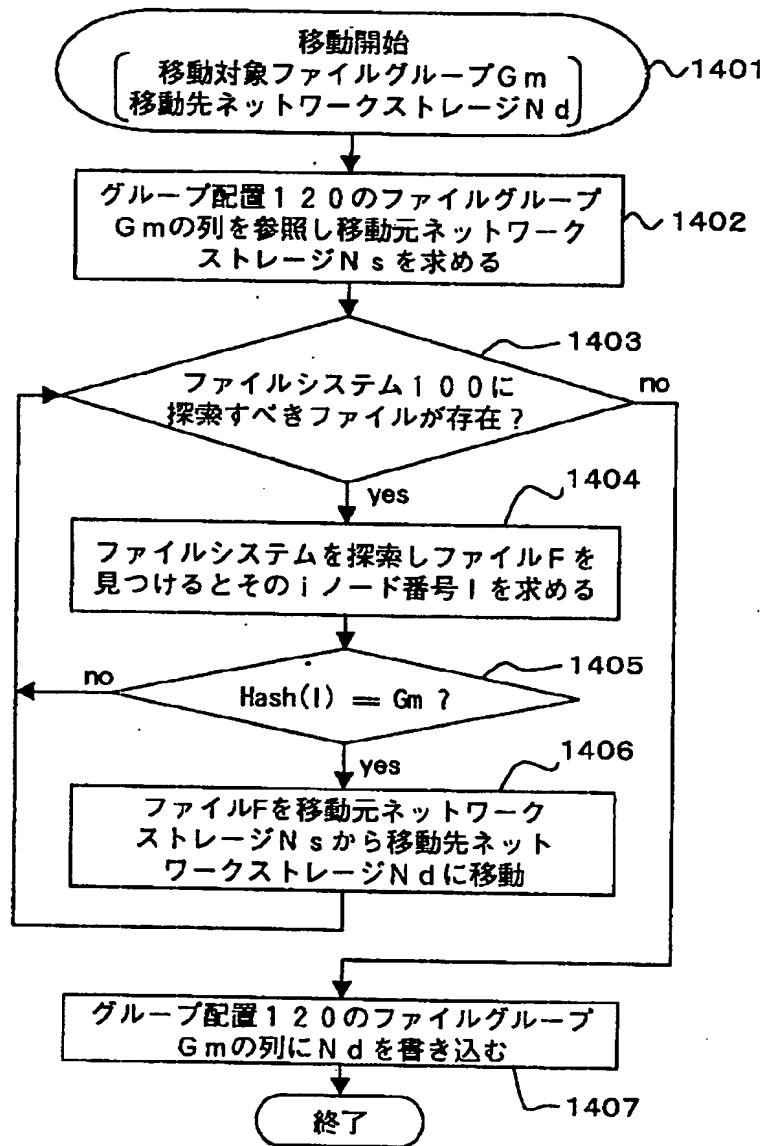
【図11】

図11



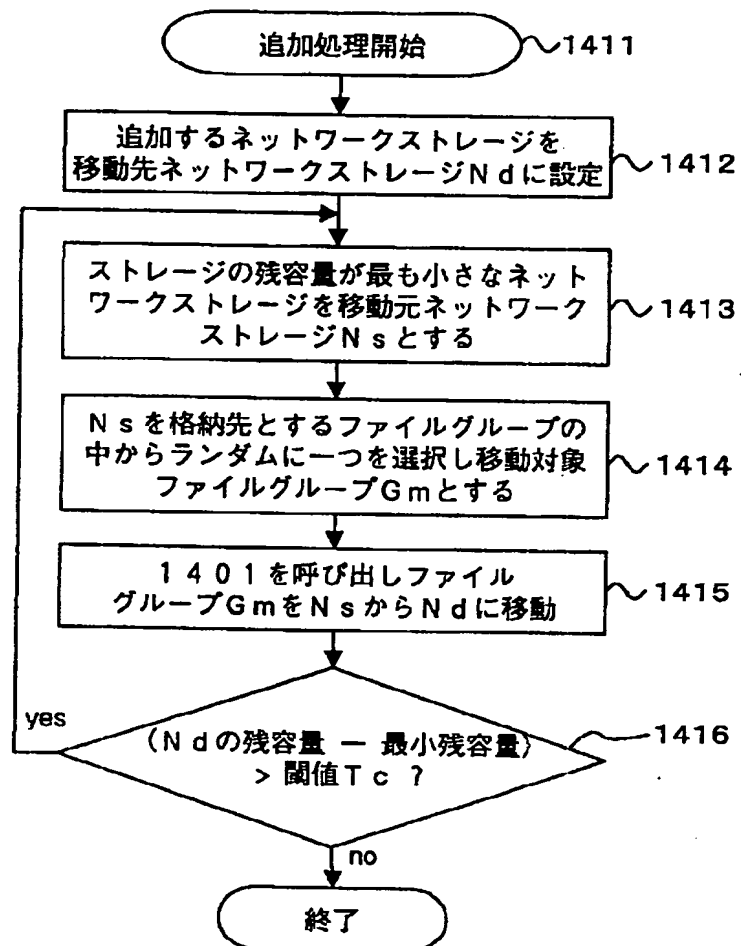
【図12】

図12



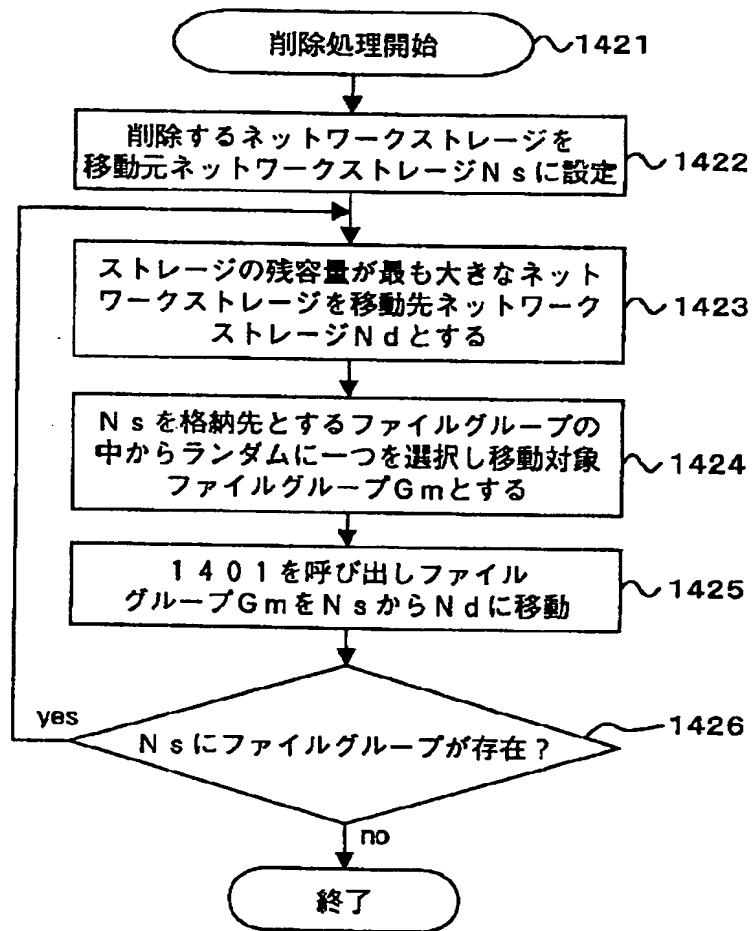
【図13】

図13



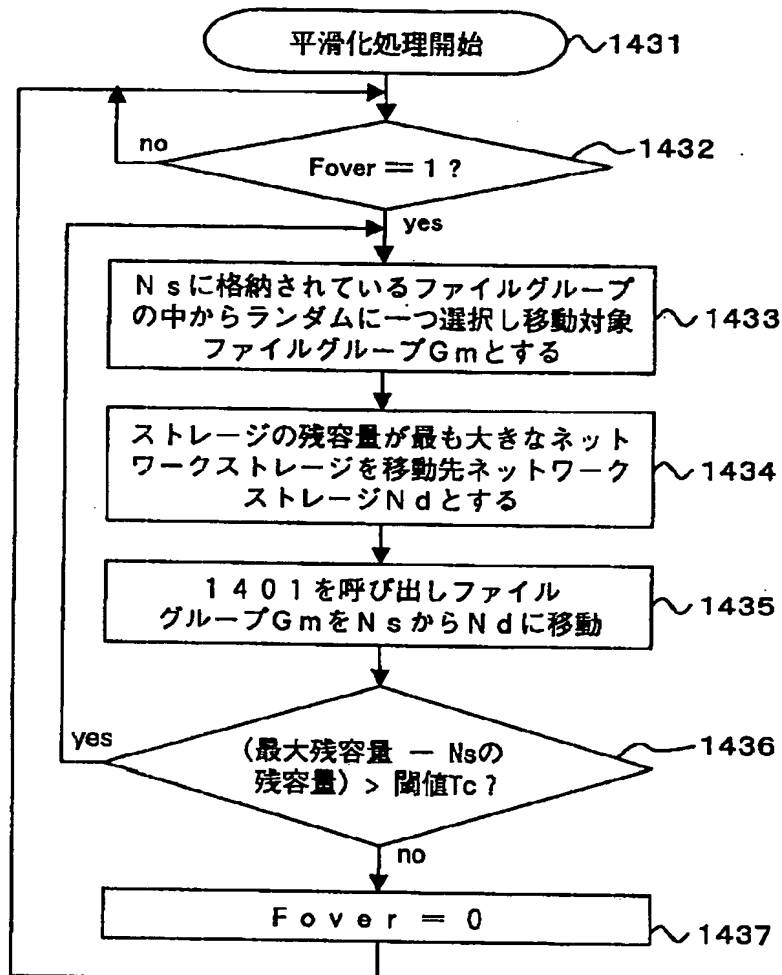
【図14】

図14

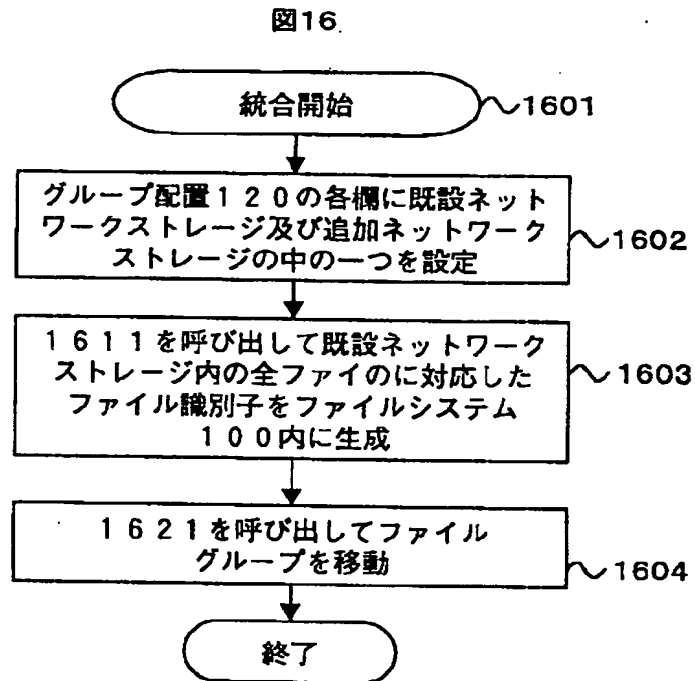


【図15】

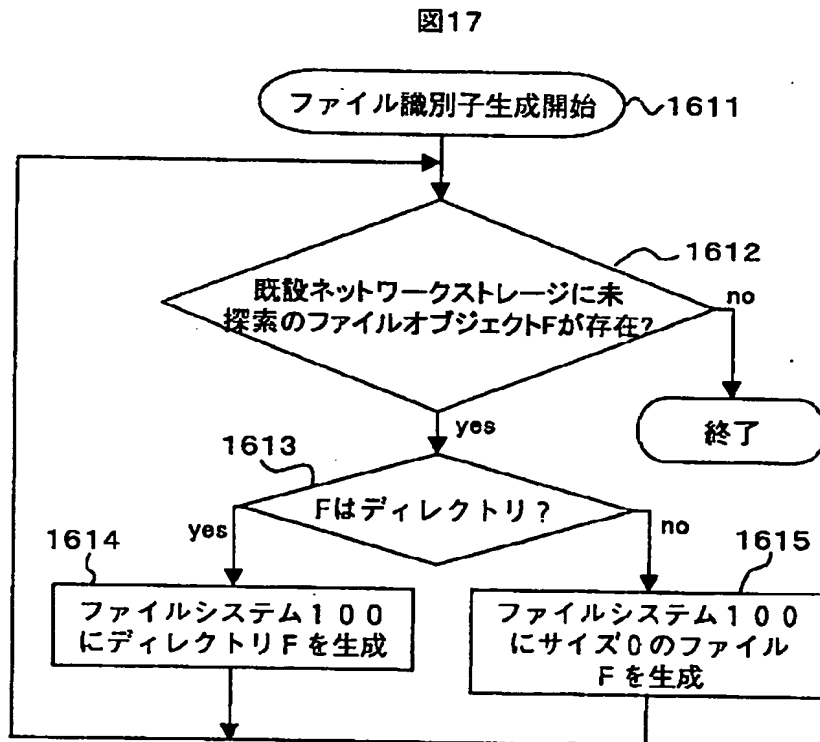
図15



【図16】

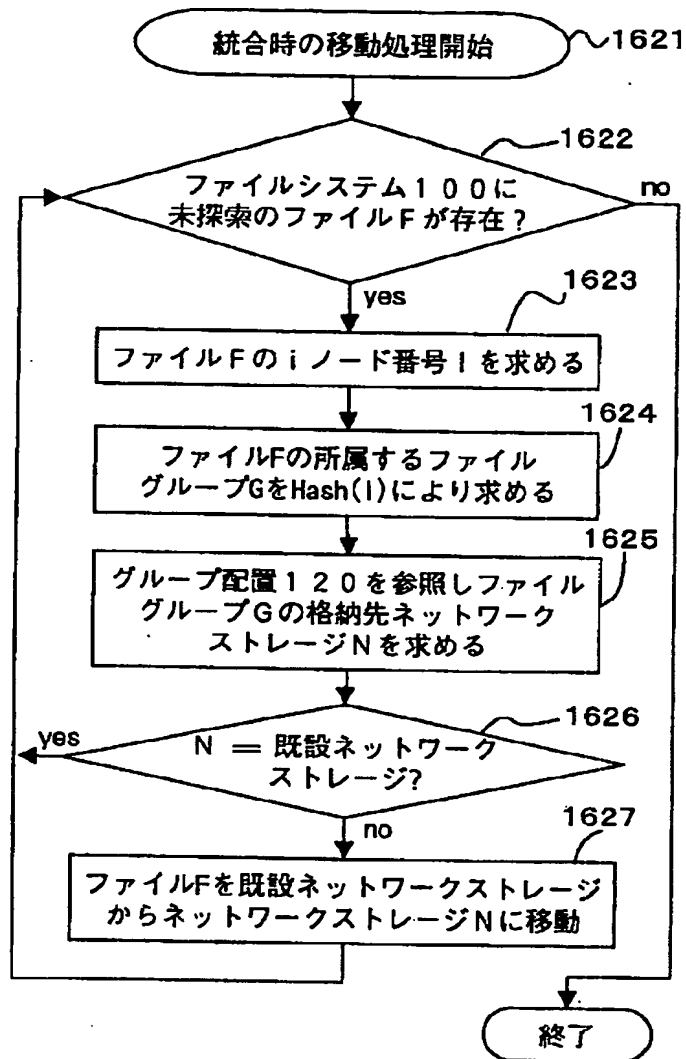


【図17】



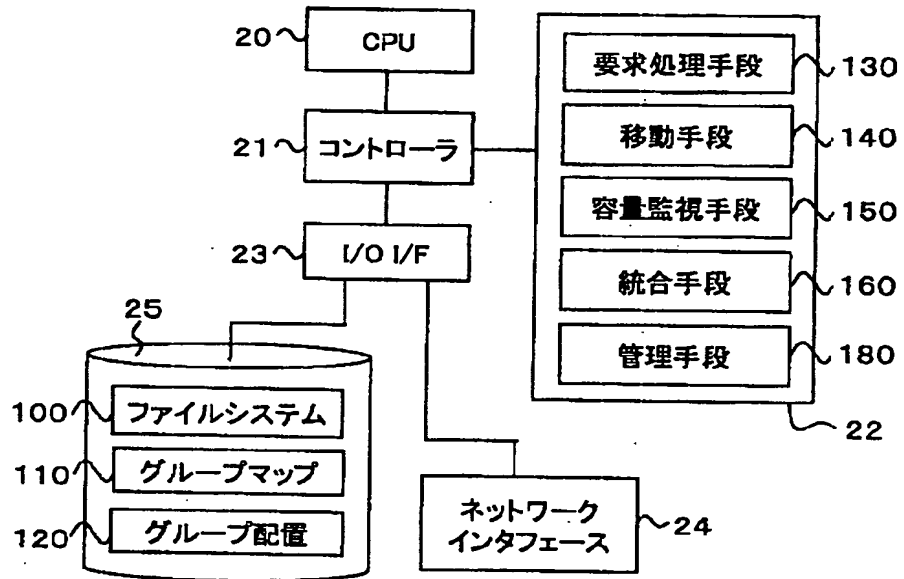
【図18】

図18

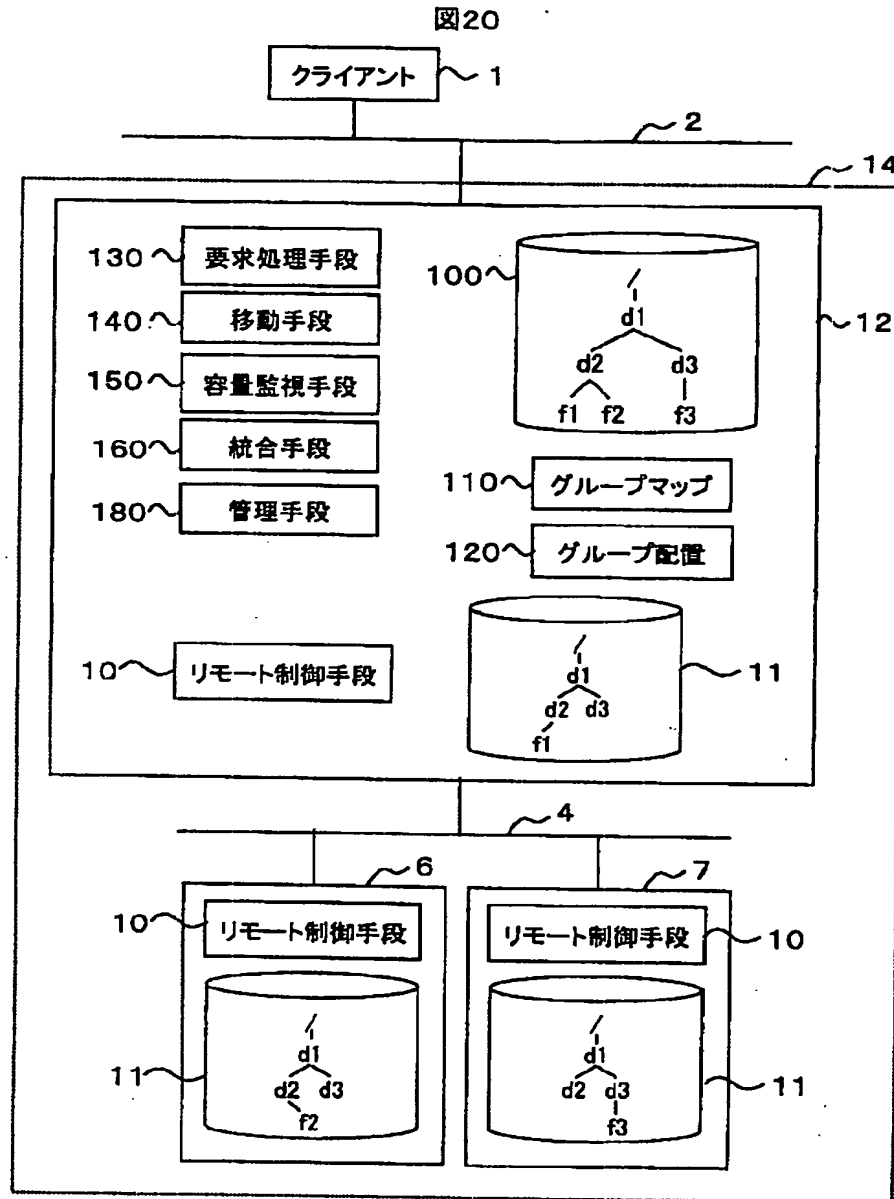


【図19】

図19

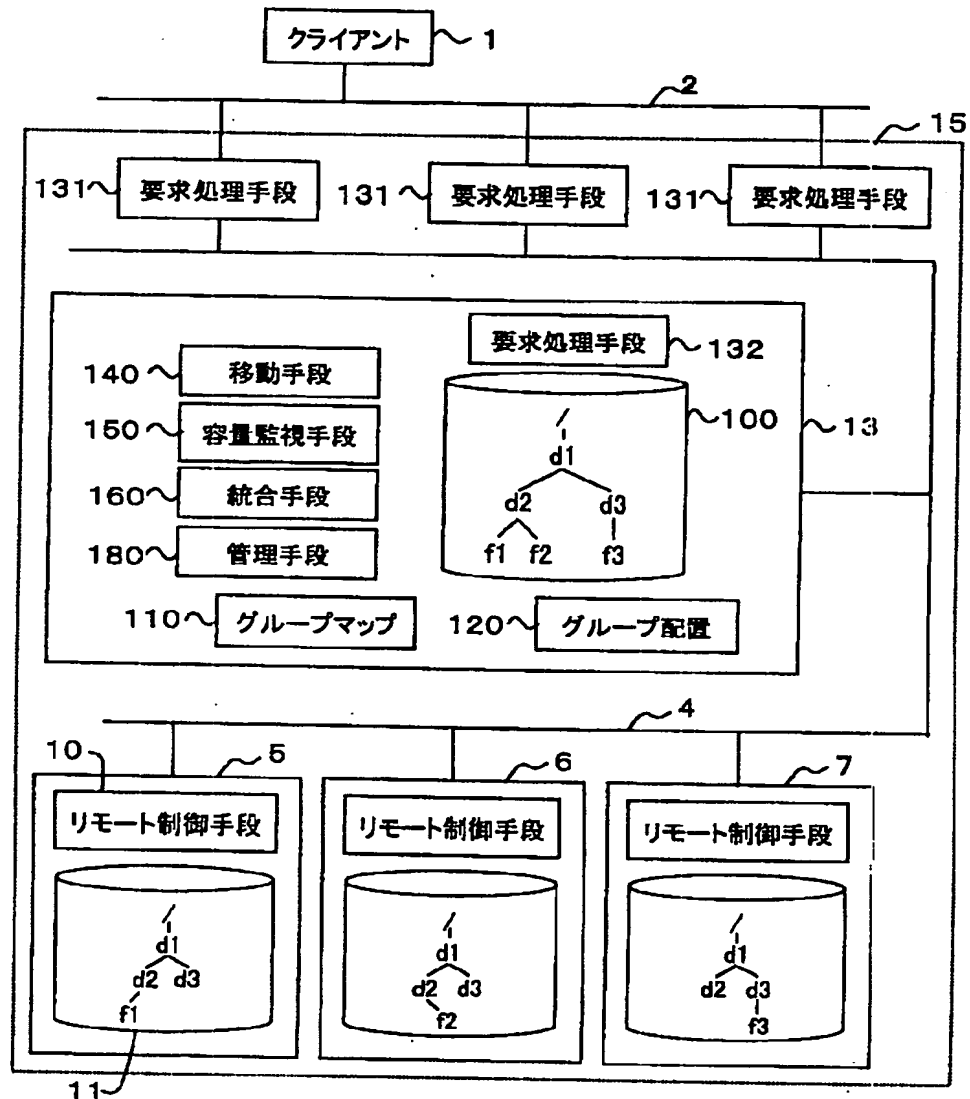


【図20】



【図21】

図21



フロントページの続き

(72)発明者 沖津 潤
東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内
(72)発明者 保田 淑子
東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内

(72)発明者 樋口 達雄
東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内
Fターム(参考) 5B065 BA01 CE01 ZA16
5B082 CA18

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ BLACK BORDERS

☒ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES

☐ FADED TEXT OR DRAWING

☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING

☐ SKEWED/SLANTED IMAGES

☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS

☐ GRAY SCALE DOCUMENTS

☒ LINES OR MARKS ON ORIGINAL DOCUMENT

☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.